BOEING

International Space Station Program

D684-10085-1
Revision B

Command & Control Software Development Plan

Type 4

August 13, 1998

Submitted to The Boeing Company
Houston, Texas
Contract No. NAS15-10000

REVISION AND HISTORY PAGE

REV.	DESCRIPTION	PUB. DATE
--	Initial Release	03/12/96
A	Revision A	07/01/98
B	Revision B	08/21/98

CONCURRENCE BY:

Boeing Prime: Kathleen Haase

PRINT NAME

/s/John Morton for KMH

SIGNATURE

NASA: David Pruett

PRINT NAME

/s/David Pruett

SIGNATURE

INTERNATIONAL SPACE STATION PROGRAM

COMMAND AND CONTROL SOFTWARE DEVELOPMENT PLAN

PREFACE

This Boeing document D684-10085-1, Command and Control Software Development Plan (SDP), describes the approach and methodology to cost-effectively manage, develop, test, control, and document the Command and Control Software (CCS) and the Node 1 Control Software (NCS) for the International Space Station (ISS). Information provided herein is submitted in accordance with the Prime Contractor SDP, D684-10017-1, the guidelines of Department of Defense Standard (DOD-STD)-2167, and DOD-STD-2167A Data Item Description (DID) tailoring as specified in the Prime Contractor Software Standards and Procedure Specification (SSPS), D684-10056-1.

KEY WORDS

Command and Control Software

Computer Software Configuration Item

Configuration Management

Flight Software

Integrated Product Team

International Space Station

Multiplexer/Demultiplexer

Node 1 Control Software

Software Development Plan

Software Engineering Environment

/s/J. W. Sherrill_____ _____
Joe Sherrill Date
Command and Control
Integrated Product Team

INTERNATIONAL SPACE STATION PROGRAM

COMMAND AND CONTROL SOFTWARE DEVELOPMENT PLAN

PREPARED BY:	_____	Boeing
	PRINT NAME	ORGN
	_____	_____
	SIGNATURE	DATE
CHECKED BY:	_____	Boeing
	PRINT NAME	ORGN
	_____	_____
	SIGNATURE	DATE
SUPERVISED BY (BOEING):	Kevin Mutz	Boeing
	PRINT NAME	ORGN
	/s/Kevin Mutz	8/14/98
	SIGNATURE	DATE
DQA:	_____	Boeing
	PRINT NAME	ORGN
	/s/Susan Davis	8/14/98
	SIGNATURE	DATE

INTERNATIONAL SPACE STATION PROGRAM

COMMAND AND CONTROL SOFTWARE DEVELOPMENT PLAN

LIST OF CHANGES

All changes to sections, tables, and figures in this document are shown below.

ENTRY DATE	CHANGE	SECTION(S)
October 4, 1996	Changed MTM to RTU	3.9
October 4, 1996	Changed MTM to RTU	F3.9.1
October 8, 1996	Typo - change “status” to “states”	5.2.2
October 8, 1996	Typo - in accordance “with”	6.2
October 9, 1996	Modified section to remove SPS & STD, vol. 1 from CCS CDR requirements	9.1
October 16, 1996	Two releases now for NCS	1.2.1.2
October 16, 1996	Clarify integration responsibility for externally provided CSCs	3.13
October 16, 1996	Two releases now for NCS	3.2.1
October 16, 1996	Added FRR, RF & TPM acronyms	9.4
October 16, 1996	Added note under schedule	3.2.1
January 21, 1998	Modified section to point to use of the C&C SSPS for content of SDFs	9.2
April 14, 1998	Removed references to SDFs throughout the document	All
April 14, 1998	Programming languages used clarification	9.1
April 27, 1998	Modified sections 9.1 and 9.2 to add sections specific to NCS and CCS	9.1.1, 9.1.2, 9.2.1 & 9.2.2

August 13, 1998	Replaced section 9.1 and 9.2 to resolve SDF and SPE issues	9.- 9.2.2
August 13, 1998	Formatting changes	All

LIST OF CHANGES - CONTINUED

ENTRY DATE	CHANGE	TABLE
October 5, 1996	Remove SUM from PDR, remove SPS from PDR (CCS)	6.0-1
April 4, 1997	Deleted SPS from table (CCS)	3.1-1
March 30, 1998	Remove S/W Product Specification and added NCS DBDD and NCS TLDD	6.0-1
August 13, 1998	Replaced Table 6.1.1-1 to resolve SDF and SPE issues	6.1.1-1

ENTRY DATE	CHANGE	FIGURE
June 19, 1998	Enlarged drawing for readability	3.1.1-2

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

This page intentionally left blank.

1. SCOPE

1.1 Identification

The Command and Control SDP establishes the policies, procedures and guidelines for the development and test of the Prime developed embedded software and its related support software. This plan applies to the CCS Computer Software Configuration Item (CSCI), the NCS CSCI, and related developmental support tools and simulations. The CCS and the NCS CSCIs will be developed and documented in accordance with DOD-STD-2167A, with appropriate tailoring as defined in the SSPS, D684-10056-1, as well as in accordance with the Prime Contractor SDP, D684-10017-01.

1.2 System Overview

The ISS system is comprised of an on-orbit facility, which is logistically supported with consumables, maintenance items, experiments, and ground facilities. The Command & Control (C&C) Multiplexer/Demultiplexer (MDM) contains CCS flight software for the top-level control functions of the ISS. The flight software contained in the C&C MDM is allocated to one CSCI (the CCS CSCI), as discussed in section 1.2.1.1. Support and test software for the CCS CSCI is also managed under this software development plan, as discussed in section 1.2.2. With the exception of the NCS CSCI, CSCIs which interface with the CCS are the responsibility of other contractors.

The Node 1 MDM contains flight software for early top-level control functions of the ISS until the C&C MDM becomes functional. The flight software contained in the Node 1 MDM is allocated to one CSCI (the NCS CSCI), as discussed in section 1.2.1.2. Support and test software for the NCS CSCI is also managed under this software development plan, as discussed in section 1.2.2. With the exception of the CCS CSCI, CSCIs which interface with the NCS are the responsibility of other contractors.

1.2.1 Flight Software

1.2.1.1 Command and Control Software CSCI

The three on-board C&C MDMs contain the CCS CSCI that provides control and coordination of designated functions for the ISS. One MDM is on line, a second is a warm standby, and the third is a cold standby. The CCS CSCI provides top level control on all control buses for data collection and distribution, and it provides for coordination of overall station functions, which include mode control, command processing, telemetry transmission, and resource management. The CCS CSCI interfaces with other flight CSCIs resident in lower level MDMs, as illustrated in figure 1.2-1.

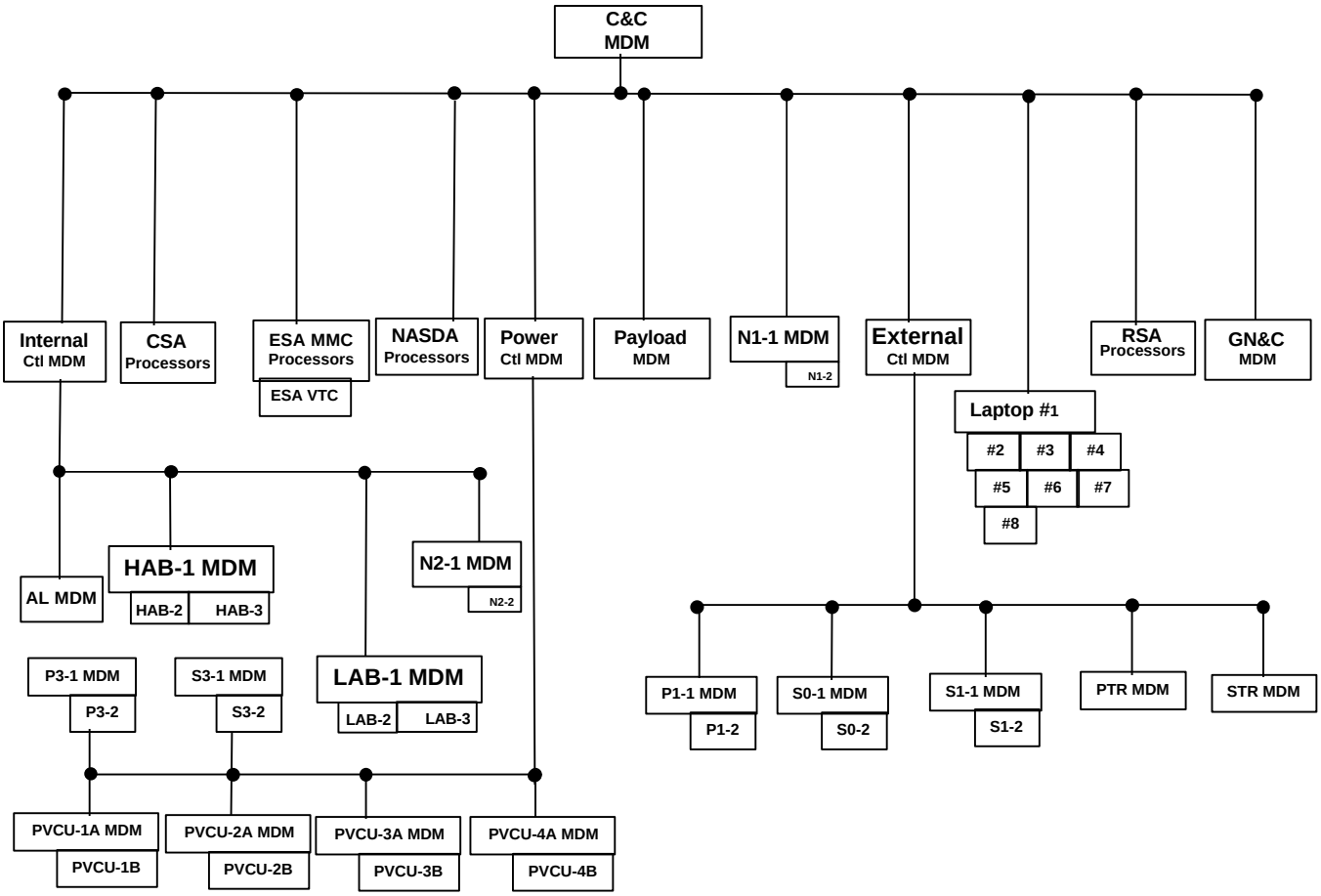


FIGURE 1.2-1 CCS TO LOWER LEVEL MDMS

The CCS CSCI is planned to be developed in five incremental releases, each with increased capability, that support the assembly and operational characteristics of the ISS. Most of the software structure and capabilities will be implemented in flight #5A. Subsequent launch assemblies are planned to require minor additions and changes to the software (each release is estimated to contain approximately 10 percent modifications to the CCS). Specifically, the release sequence, which supports the assembly and operational characteristics of the ISS, is planned as follows:

a. Release 1 of the CCS CSCI is launched with the United States Laboratory (USL) on assembly flight #5A. This is the initial release of the CSCI and provides station functionality to assembly flight #10A.

b. Release 2 of the CCS CSCI is uplinked during assembly flight #10A and provides functionality to assembly flight #1J.

c. Release 3 of the CCS CSCI is uplinked during assembly flight #1J and incorporates the additional capabilities for interfacing with the Japanese Experiment Module (JEM). This release is effective to assembly flight #1E.

d. Release 4 of the CCS CSCI is uplinked during assembly flight #1E to support the Euro-Columbus Attached Pressurized Module (APM) developed by the European Space Agency. This release is effective to assembly flight #15A.

e. Release 5 of the CCS CSCI is uplinked during assembly flight #15A and provides the additional capabilities required to integrate the United States Habitation (HAB) module into the ISS. This release is effective through and after the completion of the ISS assembly.

Changes to the contents of a release or the release sequence may be required to fulfill the evolving requirements of the ISS. Formal Qualification Test (FQT) will be performed for each software release (as shown in section 3.2.1.2).

1.2.1.2 Node 1 Control Software CSCI

The two on-board Node 1 MDMs contain the NCS CSCI that provides control and coordination of designated functions for the ISS. The NCS CSCI provides top level control on all data buses connected to the Node 1 MDM for data collection and distribution, and it provides for coordination of overall station functions, which include command processing, telemetry transmission, and resource management from assembly flight #2A until the CCS takes over these functions during assembly flight #5A. In addition, the NCS provides a portion of the Thermal Control System (TCS) functions and a portion of the Environmental Control and Life Support System (ECLSS) functions throughout the life of the ISS. Both MDMs are on line, with one MDM providing the primary control and monitoring for some of the Node 1 functions with the other MDM providing the secondary control and monitoring for those functions. At the same time each MDM provides the primary control and monitoring of a unique set of sensors and effectors. The NCS CSCI interfaces with other flight CSCIs resident in various MDMs.

The NCS CSCI is planned to be developed in two incremental releases, each with increased capability, that support the assembly and operational characteristics of the ISS. Most of the software structure and capabilities will be implemented in Flight 2A. The subsequent launch assembly that is planned requires minor additions and changes to the software, and will be implemented in Flight 4A. The release sequence can be summarized as follows:

a. Release 1 of the NCS CSCI is on Assembly Flight 2A. This is the initial release of the CSCI and provides station functionality to assembly Flight 4A.

b. Release 2 of the NCS CSCI is uplinked during assembly Flight 4A and provides functionality to assembly complete.

1.2.2 Simulation Software

1.2.2.1 Command and Control Environment Simulation Software

The Command and Control Environment Simulation (CES) software resides in the platform used to perform both the informal and formal testing of CCS CSCI. The CES consists of interface simulations of each Military (MIL)-STD-1553 Remote Terminal (RT) and devices connected via Recommended Standard (RS)-485 that interface directly to the CCS CSCI, data recording, reduction and analysis software, and fault insertion software.

The CES will be placed under Software Configuration Management control but will not be delivered as part of the Flight Software Code.

An adapted version of the CES is provided to the Software Verification Facility (SVF) for use in performing Command & Data Handling (C&DH) level verification testing. As a minimum, the CES will be adapted to the SVF platform.

1.2.2.2 Node 1 Control Environment Simulation Software

The Node 1 Control Environment Simulation (NES) software resides in the platform used to perform both the informal and formal testing of NCS CSCI. The NES consists of interface simulations of each MIL-STD-1553 RT, sensor and effector that interface directly to the NCS CSCI, data recording, reduction and analysis software, and fault insertion software.

The NES will be placed under Software Configuration Management control but will not be delivered as part of the Flight Software Code.

An adapted version of the NES is provided to the SVF for use in performing C&DH level verification testing. As a minimum, the NES will be adapted to the SVF platform.

1.3 Document Overview

This document is designed to provide the reader with a basic understanding of how the software developers intend to develop the CCS CSCI and the NCS CSCI software and manage that development process toward ensuring a quality product upon completion. This document identifies the software to be developed, the documentation to be produced, the development methodology to be followed, the management approach to be used in directing, controlling, and supporting the software development, and the quality assurance provisions employed in monitoring this development.

This plan complies with DOD-STD-2167A, DID DI-MCCR-80030A, as tailored by the Prime Contractor SSPS, D684-10056-1. Section 2, *Referenced Documents*, provides identification for documents referenced in this SDP. Section 3, *Software Development Management*, describes planning associated with the software development management activities. Section 4, *Software Engineering*, describes the techniques, methods, and processes to be used in software development. Section 5, *Formal Qualification Testing*, describes planning associated with software formal testing activities. Section 6, *Software Product Evaluations*, describes planning associated with software product evaluation activities. Section 7, *Software Configuration Management*, describes software configuration management activities. Section 8, *Other Software Development Functions*, discusses any other functions involved in the software development effort not previously addressed. Section 9, *Notes*, provides a list of acronyms and glossary.

1.4 Relationship to Other Plans

This SDP is compatible with and subordinate to the Prime Contractor SDP, D684-10017-1. Relationships to higher level plans are as defined in the Prime Contractor SDP, section 1.4. It is in compliance with the Prime SDP except as noted herein. All exceptions must be approved by the Prime and will be documented in section 9.1. Upon approval by the Prime, the SDP will be placed under

Software Configuration Management (SCM) control. Any future changes must be approved by the Prime.

In the event of a conflict between this document and higher level documents, the higher level documents take precedence.

This SDP is compatible with and provides the lower level details of the processes defined in the C&C Integrated Product Team (IPT) Team Execution Plan (TEP), D684-10093. The TEP also provides details not contained within this document such as the Statement Of Work, metrics reporting, scope of work, etc. The C&C IPT works closely with many ISS teams throughout the development of the flight software products. Key interfacing teams include Station Management and Control (SMC) and corresponding subsystem teams as well as other C&DH Teams. These relationships are described in more detail in the C&C TEP.

The C&C SSPS forms a part of this SDP, in that it contains the details associated with the software development techniques and methodologies, the coding and design standards, the informal and formal verification standards and processes, the configuration management processes, and the conduct and review criteria for the in-process reviews to be used to develop the NCS, CCS, NES, and CES CSCIs. This SDP defines the approach and methodologies applied for the defined activities.

The C&C Software IPT Risk Management Plan (RMP) forms a part of this SDP, in that it contains the details associated with risk management for the C&C IPT. The SDP defines the approach and methodologies applied for risk management.

This page intentionally left blank.

2. REFERENCED DOCUMENTS

This section lists all documents referenced in this SDP.

MILITARY STANDARDS

DOD-STD-2167 Military Standard, Defense System Software Development

DOD-STD-2167A Military Standard, Defense System Software Development

MIL-STD-1521B Technical Reviews and Audits for Systems, Equipment, and Computer Software 4 June 1985

NASA

JSC 2410.11	AIS Security Manual
SP-M-505	Prime Item Development Specification for the Enhanced Space Station Multiplexer/ Demultiplexer
SSP 41170	Space Station Configuration Management Requirements
SSP 50010	Space Station Documentation Requirements, Standards and Guidelines
SSP 50123	Configuration Management Handbook

BOEING

D684-10054-1	Prime Risk Management Plan
D684-10017-1	Prime Contractor Software Development Plan
D684-10056-1	Prime Contractor Software Standards and Procedures Specification
D684-10700-1	Prime Safety and Mission Assurance (S&MA) Plan
D684-TBD-1	C&C Software Integrated Product Team (C&C IPT) Risk Management Plan
D684-10093-1	C&C Software Integrated Product Team (C&C IPT) Team Execution Plan
D684-10189-01	Data Base Design Document for the Node 1 Control Software
D684-10193-01	Software Top Level Design Document for the Command and Control Software

D684-10194-01	Data Base Design Document for the Command and Control Software
D684-10195-01	Software Top Level Design Document for the Node 1 Control Software
S684-10131	C&C MDM CSCI Software Requirements Specification
S684-10143	SMC End Item Specification
S684-10171	Node 1 MDM Application End Item Specification
S684-10174	NCS MDM CSCI Software Requirements Specification
S684-10186	Software Standards and Procedures Specification for the Command and Control Integrated Product Team
SDIL - 003	SDIL Configuration Management Handbook
SSP 41175	SMC To ISS Software ICD

3. SOFTWARE DEVELOPMENT MANAGEMENT

3.1 Project Organization and Resources

This section describes the project organization and resources to be applied to the software development activity. The ISS program is organized into IPTs. Each team is responsible for the development of a product or set of products within the system. These products may be hardware, software or documentation. An IPT may have sub-IPTs, which take responsibility for a subset of the products and are composed of subsets of the personnel.

The CCS, NCS and related support software will be designed, implemented and tested by the C&C Software IPT. The C&C Software IPT is one of several sub teams under the C&DH IPT. Figure 3.1-1, C&DH IPT Organization, shows the relationship between the C&C Software IPT and the C&DH IPTs and Analysis and Integration Teams (AITs). The C&C IPT reports directly to the C&DH IPT. Flight software products of the C&C IPT are delivered to the Software CM IPT within the C&DH IPT. Details of the C&DH IPT organization may be found in the C&DH IPT TEP. Details of the C&C Software IPT relationships to other organizations may be found in the C&C Software IPT TEP. The requirements for the CCS and NCS will be developed and delivered by the SMC AIT. The SMC AIT is separate from the C&DH IPT and reports to the Subsystem Architecture AIT. The SMC AIT TEP, describes SMC activities and responsibilities in detail.

Table 3.1-1, Software Products and Responsibilities, shows the products produced by the C&C Software IPT, including the software CSCIs and the associated documents. For each product, the IPT with the primary development responsibility is indicated, along with IPTs responsible for review. Also for each product is an indication of whether the product is controlled via SCM or via engineering control. If a product is controlled by SCM, Table 3.1-1 defines the time period in which the product is placed under control. Software documents will be produced to capture the evolving and as-built description of the CCS, NCS, CES, and NES, with final as-built versions released through the normal engineering release process. The documents will be produced to aid in the configuration management and maintenance of the software product. The format and contents of the documents except for the Top Level Design Document (TLDD) and the Database Design Document (DBDD) will conform to SSP 50010 and DOD-STD-2167A as tailored per the Prime SSPS. The TLDD and the DBDD will conform to SSP 50010 and DOD-STD-2167. The TLDD and the DBDD will replace the use of the Software Design Document (SDD). (See section 9.1 for rationale on the use of the TLDD and DBDD.)

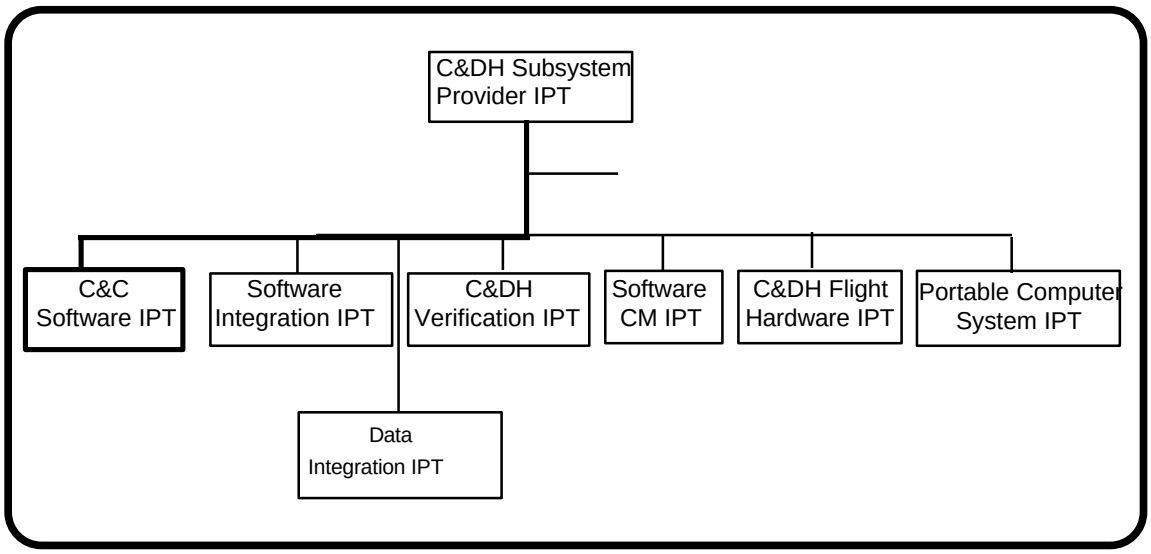


FIGURE 3.1-1 C&DH IPT ORGANIZATION

TABLE 3.1-1 SOFTWARE PRODUCTS AND RESPONSIBILITIES (PAGE 1 OF 2)

Software Product	Station Management & Control AIT	Flight Software Integration IPT	C&C Software IPT	SCM Control
Software Development Plan (SDP)	Secondary	Review	Primary	PDR
Software Standards and Procedures Specification (SSPS)		Review	Primary	PDR
CCS Computer Software Configuration Item (CSCI)			Primary, Test	TRR
CCS Software Requirements Specification (SRS)	Primary	Review	Review	PDR
CCS Top Level Design Document (TLDD)	Review	Review	Primary	No
CCS Database Design Document (DBDD)	Review	Review	Primary	No
CCS Software Test Plan (STP)	Review	Review	Primary	PDR
CCS Software Test Description (STD)	Review	Review	Primary	Vol I CDR Vol II TRR
CCS Software Test Results (STR)	Review	Review	Primary	FQT
CCS Version Description Document (VDD)			Primary	TRR
CCS Software User's Manual (SUM)		Review	Primary	TRR
NCS Computer Software Configuration Item (CSCI)			Primary, Test	TRR
NCS Software Requirements Specification (SRS)	Primary	Review	Review	PDR
NCS Top Level Design Document (TLDD)	Review	Review	Primary	No
NCS Database Design Document (DBDD)	Review	Review	Primary	No
NCS Software Test Plan (STP)	Review	Review	Primary	PDR
NCS Software Test Description (STD)	Review	Review	Primary	Vol I CDR Vol II TRR
NCS Software Test Results (STR)	Review	Review	Primary	FQT
NCS Version Description Document (VDD)			Primary	TRR
NCS Software User's Manual (SUM)		Review	Primary	TRR

TABLE 3.1-1 SOFTWARE PRODUCTS AND RESPONSIBILITIES (PAGE 2 OF 2)

Software Product	Station Management & Control AIT	Flight Software Integration IPT	C&C Software IPT	SCM Control
NES Computer Software Configuration Item (CSCI)			Primary	TRR
NES Software Requirements Specification (SRS)	Review		Primary	PDR
NES Top Level Design Document (TLDD)			Primary	No
NES Database Design Document (DBDD)			Primary	No
NES Version Description Document (VDD)			Primary	TRR
NES Software User's Manual (SUM)			Primary	No
CES Computer Software Configuration Item (CSCI)			Primary	TRR
CES Software Requirements Specification (SRS)	Review		Primary	PDR
CES Top Level Design Document (TLDD)			Primary	No
CES Database Design Document (DBDD)			Primary	No
CES Version Description Document (VDD)			Primary	TRR
CES Software User's Manual (SUM)			Primary	No
Team Execution Plan (TEP) *			Primary	No
Risk Management Plan (RMP) *			Primary	No

* note: The TEP and RMP will be reviewed by the C&DH IPT

3.1.1 Contractor Facilities

The CCS CSCI and the NCS CSCI are developed at the Sonny Carter Training Facility (SCTF) in Houston, Texas. The SCTF is a government furnished facility. The Software Development and Integration Lab (SDIL) is contained within the SCTF and consists of the Prime Software Production Facility (PSPF), the Mission Build Facility (MBF), and the SVF. A graphical representation of the SDIL components and their relationships and product flow is show in Figure 3.1.1-1, SDIL Components.

The PSPF is the primary software development resource needed for the development of the CCS CSCI, NCS CSCI, CES CSCI, and the NES CSCI. The PSPF provides the Boeing Prime suite of tools and the Software Engineering Environment (SEE) tools. The MBF is the primary repository of all configuration controlled material relating to the C&DH system. The primary interface between the MBF and C&C Software IPT is that the MBF will be the primary repository for the software products produced by the C&C Software IPT and will be the supplier of command and telemetry information pertinent to the development of the CCS and NCS. The SVF provides the Boeing Prime suite of tools for the development and conduct of Stage Verification Tests. The primary interface between the SVF and C&C Software IPT is that the C&C Software IPT will provide the CCS, NCS, CES, and NES code, though the MBF, and also provide assistance in anomaly resolution at stage verification.

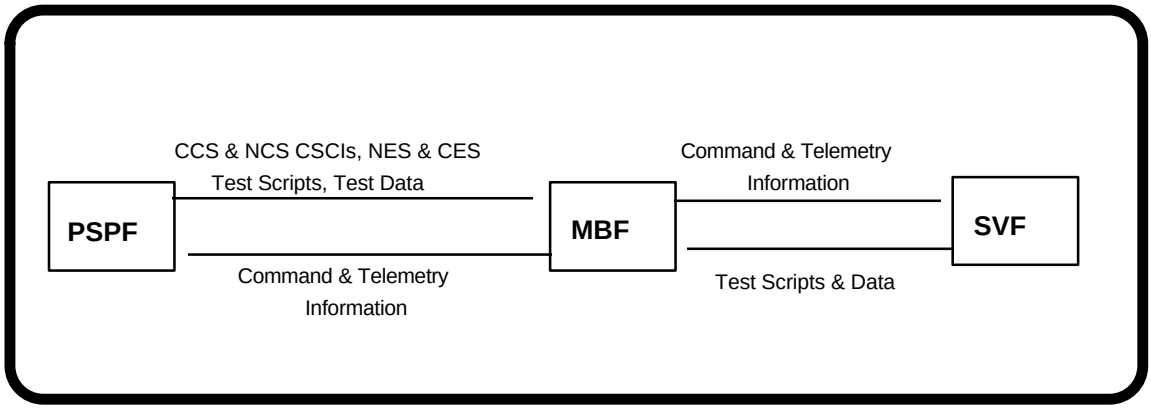


FIGURE 3.1.1-1 SDIL COMPONENTS

The development functions supported by the PSPF, and their allocation to hardware are shown in Figure 3.1.1-2, PSPF - Functional Allocation. A description of the SEE and its hardware, software, and firmware is contained in section 4.1.3 of this plan. Responsibility for the shared development facility has been assigned to the Prime SDIL IPT/AIT.

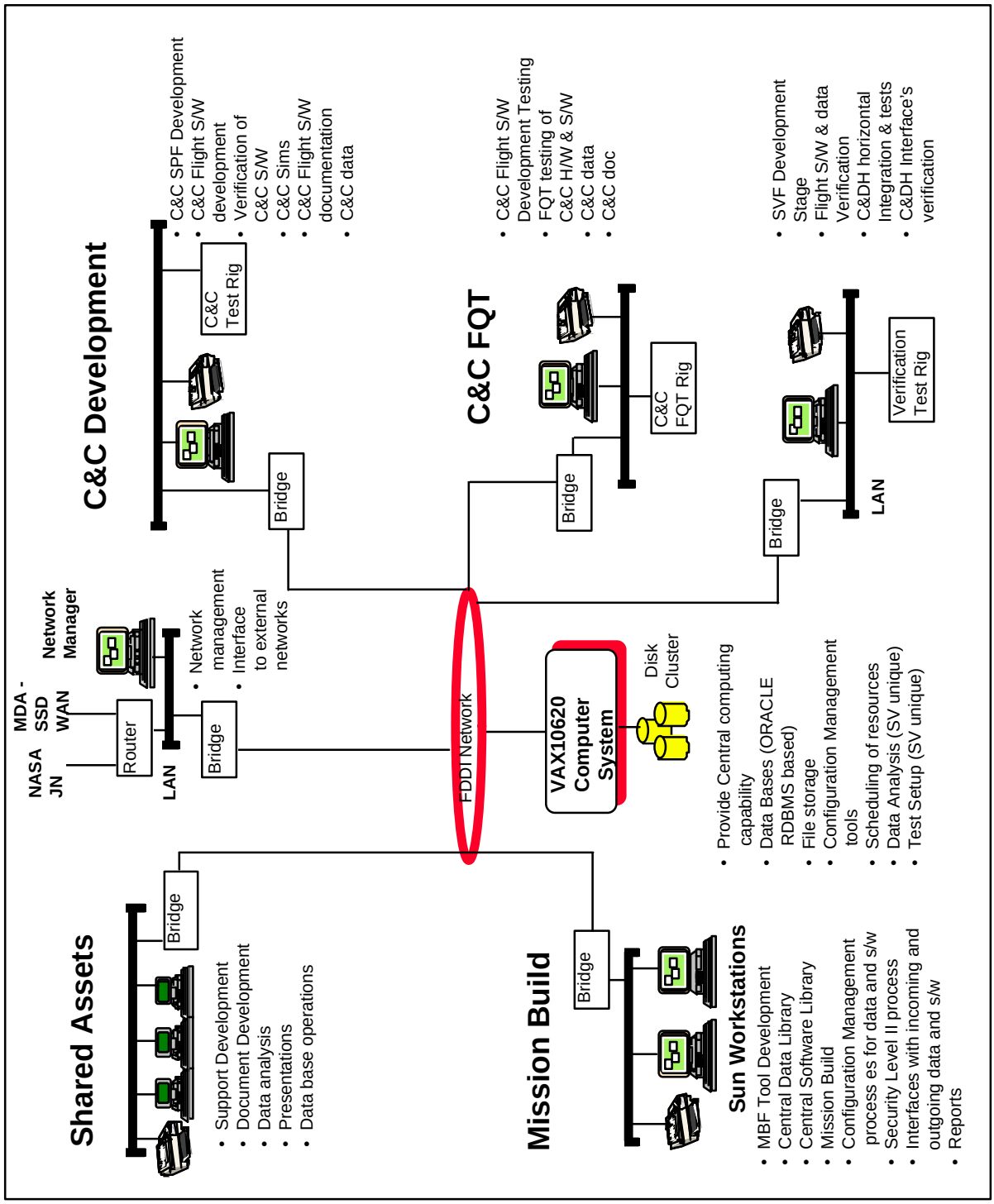


FIGURE 3.1.1-2 PSPF – FUNCTIONAL ALLOCATION

3.1.2 Government Furnished Equipment, Software, and Services

The National Aeronautics and Space Administration (NASA) will provide Timeliner (which is a procedure execution controller being developed by Draper Labs) as a Government Furnished Equipment (GFE) item. Timeliner will be integrated with the CCS and included as part of the overall CSCI.

As a result of development facility planning, a list of hardware, software, and other resources required will be compiled. A copy of the list of required items and their sources, (i.e., through specific commercial vendors), supplied by the prime contractor or the government, will be maintained by the SDIL IPT/AIT.

3.1.3 Organizational Structure

The C&C Software IPT organization provides the CCS CSCI for the C&C MDMs and the NCS CSCI for the Node 1 MDMs. The C&C Software IPT supports the definition of the CCS and NCS software requirements. As a software provider, the C&C Software IPT designs, develops, verifies, and delivers the CCS and the NCS. The C&C Software IPT also provides simulations, test support, and services to other organizations.

The C&C Software IPT is comprised of the Software Development (Flight Software and Simulation) Group, the Integration and Test Group, and the System Engineering Group as shown in Figure 3.1.3 -1, C&C Software IPT Organization. Details of the C&C Software IPT organization may be found in the C&C IPT TEP.

The C&C Software IPT is jointly lead by a Boeing and a NASA manager. The C&C Software IPT meets on a regular basis to coordinate plans, provide development status, and discuss/resolve issues.

The SMC AIT produces and maintains the CCS and NCS software requirement specifications, S684-10131 and S684-10174. These requirements are developed and derived from requirements allocated from higher level specifications, the SMC End Item Specification, S684-10143 and the Node 1 MDM Application End Item Specification, S684-10171. The details of the SMC AIT organizational structure is described in the SMC TEP.

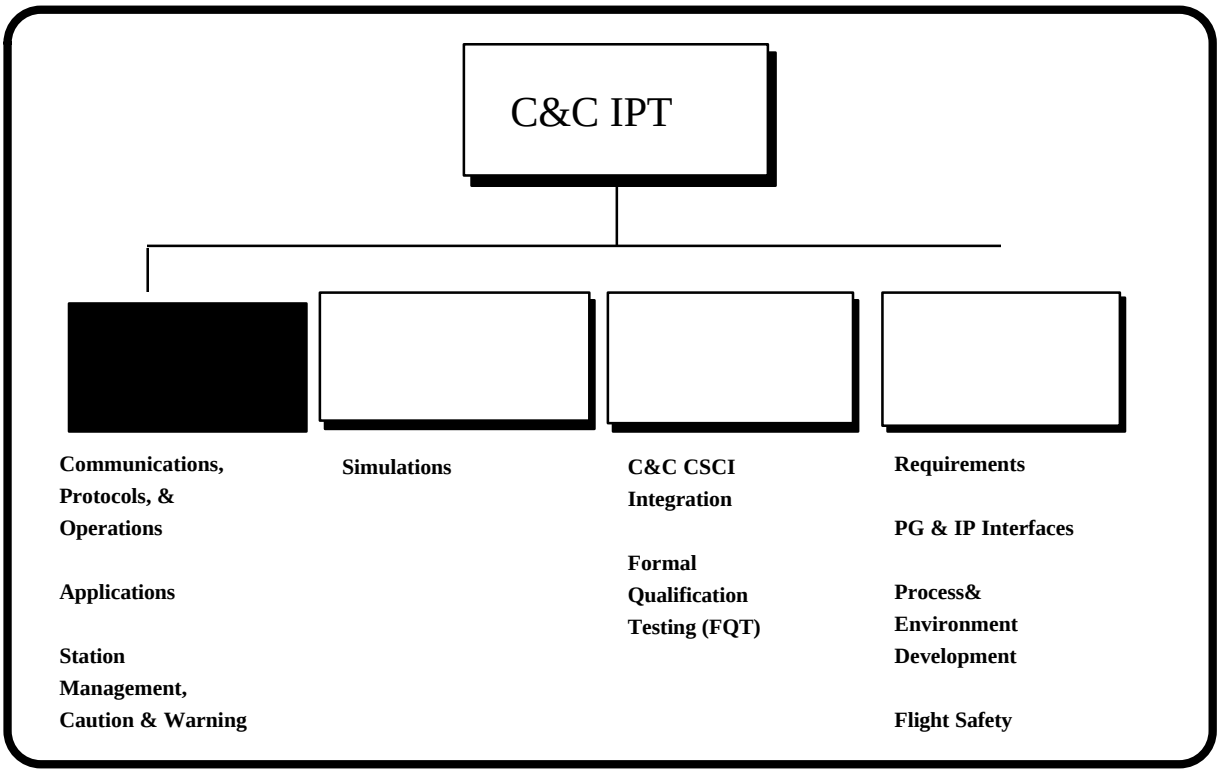


FIGURE 3.1.3-1 C&C SOFTWARE IPT ORGANIZATION

The C&C Software IPT will receive software from and provide data and programs to the Mission Build Operations. The C&C Software IPT will maintain its own databases and software configuration management processes (which will maintain similarity and compatibility with mission build operations as practical).

The C&C Flight Software Development Group has the responsibility to design and develop the NCS and CCS. This group is responsible for the system level and the detailed design of CCS and NCS. This group will test their developed code at the computer software unit (CSU) and computer software component (CSC) levels. The C&C Flight Software Development Group will test the NCS and CCS CSCIs for compliance with the design. The C&C Flight Software Development Group will also integrate externally provided CSCs (e.g., MDM Utilities and Timeliner) into the CCS or NCS CSCI. The interface to these externally provided CSCs will also be tested for compliance with the design. The C&C Flight Software Development Group provides requirements definition support to the SMC AIT.

The C&C Simulation Software Development Group has the responsibility for the development of the NES and the CES for CSCI testing of the NCS and the CCS. The NES and CES will also be adapted by the C&C Simulation Software Development Group for use in C&DH system level verification testing at the SVF. The CES and NES versions used to support FQT CSCI testing of the CCS and NCS will be placed under Software Configuration Management control but will not be delivered as part of the Flight Software Code. The CES and NES versions used in the SVF for system level verification will be placed under Software Configuration Management control but will not be delivered as part of the Flight Software Code. A goal is to use other development organizations' (including all Product Groups (PGs)) simulation software with minimum conversion.

The C&C Integration and Test Group has the responsibility to plan the FQT, write the FQT procedures, run the FQT, and document the FQT results to ensure compliance of the NCS and CCS CSCI to the requirements documentation. This group is independent from the C&C Flight Software Development Group. This group will also provide support for the C&DH system level verification tests and for the PG-1 and PG-3 integration activities.

The C&C Systems Engineering (SE) Group supports the SMC AIT in the definition of functional, performance, and interface requirements for the NCS and CCS. They act as the C&C Software IPT technical point of contact to the ISS IPTs, PGs, and International Partners (IPs). The SE Group defines the processes and capabilities needed for development of the NCS and CCS. The SE Group represents the C&C Software IPT at the Safety & Mission Assurance (S&MA) reviews leading to Flight Readiness Review.

A separate SCM group (with personnel assigned to the C&C Software IPT) will maintain and control software build releases, databases, and documentation for design, development, and test. This group will maintain a problem reporting system and provide controlled software build releases as required for the C&C Software IPT. SCM procedures may be a subset of the C&DH configuration management procedures.

3.1.4 Personnel

The C&C Software IPT consists mainly of software design engineers and software test engineers. The IPT consists of Boeing Prime, NASA, and subcontractor personnel directly assigned to the C&C Software IPT working as a single development and test team. Personnel requirements, skills, and staffing levels are maintained by C&C Software IPT management.

The SMC AIT organization consists of system and software engineers. The SMC AIT also brings together Boeing Prime, NASA, and subcontractor personnel.

Software Quality, System Safety, and Software Configuration Management engineers, although not directly part of the C&C Software IPT and the SMC AIT, participate in the development through active participation in the overall development process. Participation is coordinated by inclusion of these disciplines in the planning activities and involvement associated with each In-Process Review (IPR).

3.2 Schedule and Milestones

3.2.1 Activities

The CCS and NCS software is developed and tested in a phased engineering approach, broken into three major activities; flight software development, flight software qualification, and simulation software development. This approach will be used for each of the five planned releases for CCS and CES, and the two releases for NCS and NES. Each of the major activities is described below, and the relationship of each activity to others is described in section 3.2.2.

The Tier III schedule is contained in the C&DH Master Schedule and is used by the C&DH IPT to monitor progress. This schedule will be maintained and controlled by the C&DH program planning and control organization. The Tier III schedule shows each development phase and the major milestones such as formal reviews. The Tier III schedule includes products to be delivered to the C&C Software IPT from external groups and products to be delivered by the C&C Software IPT to external groups. This schedule also shows the early releases of CCS and NCS necessary to support program milestones. The Tier III schedule is used by the C&C Software IPT manager to plan and monitor software development activities necessary to support other teams.

The C&C Software IPT manager decomposes the Tier III schedule into lower-level schedules for the software development activities. These schedules show the activities required for the development of each CSC including associated documentation and integration of the CSCs into the CSCI. These schedules will be maintained and controlled by the C&C Software IPT manager. The detailed schedules are used for maintaining schedule status information and are the basis for assigning tasks to software development personnel and for monitoring work progress.

The C&C Software Development IPT schedule for Stage 2A and 5A is shown in Figure 3.2.1-1.

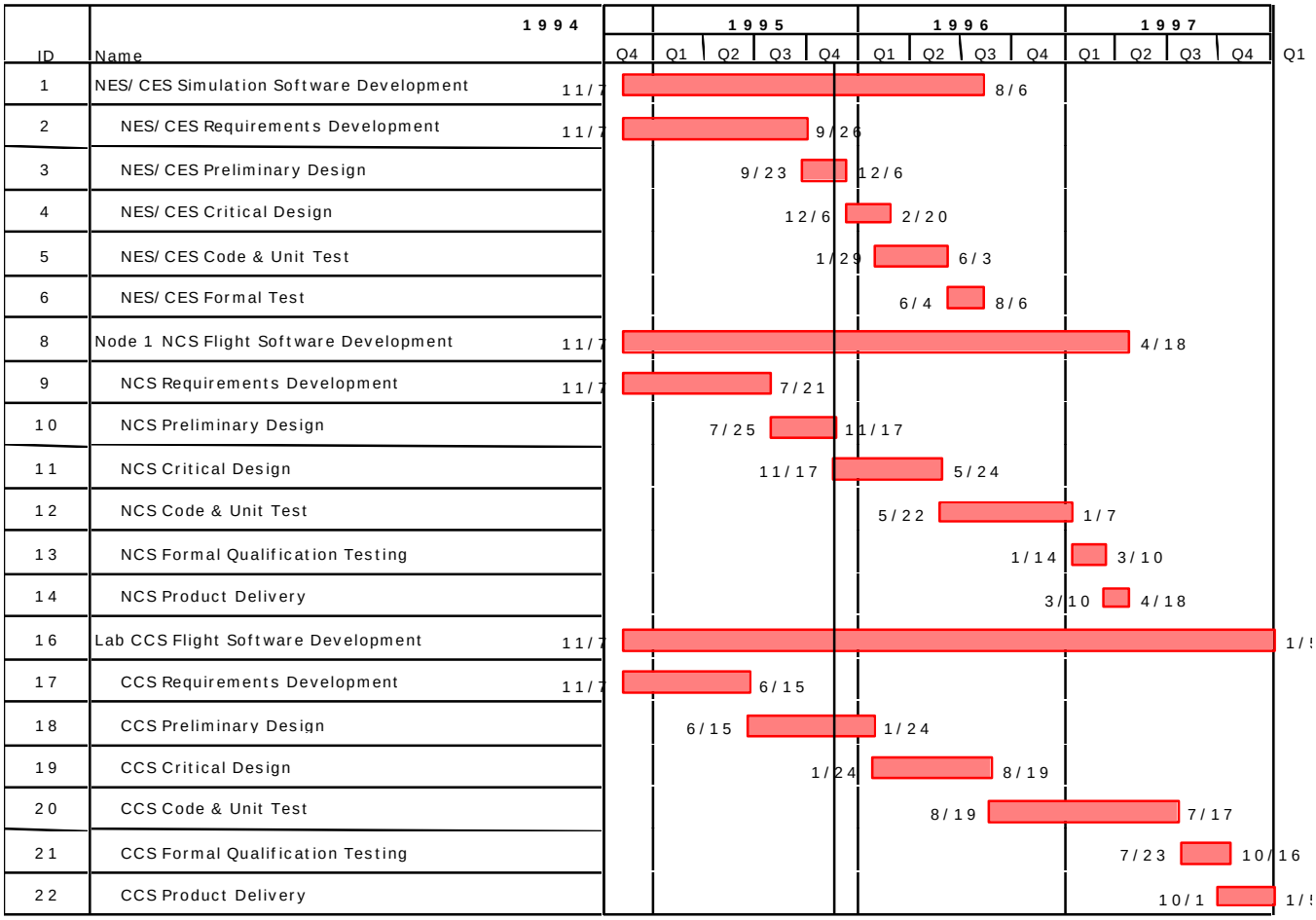
3.2.1.1 Flight Software Development

The CCS and NCS flight software development process is a phased development process consisting of requirements analysis, preliminary design, detailed design and code/unit test/verification phases.

The requirements analysis phase will produce a Software Requirements Specification (SRS) to be presented at the Software Requirements Review (SRR). During this phase breadboard code will be developed to allow initial hardware/software integration, substantiate sizing and timing estimates, and provide functional capability for preliminary design phase integration. Thorough CSU and CSC testing will not be performed in this phase. Completion of the requirements analysis phase allows initiation of the preliminary design phase.

Command and Control Software Development IPT

Stage 2A and 5A Software Development



Note: This schedule is provided as a guide for understanding flight software development. Updates to the schedule are provided at all the reviews as well as C&DH Management Meetings.

FIGURE 3.2.1-1 C&C SOFTWARE DEVELOPMENT IPT SCHEDULE - STAGE 2A AND 5A

During the preliminary design phase the high level design of the software will be documented in a TLDD and a DBDD. The preliminary design will be presented at the Preliminary Design Review (PDR). This review will be held to assure compatibility of the software design with its requirements and other subsystems. During this phase brassboard code will be developed to add functional capabilities to input and output CSCI external data and to substantiate sizing, timing, and interface loading estimates. Thorough CSU and CSC testing will not be performed in this phase.

The detailed design phase will follow the completion of the Preliminary design phase. The detailed design will be documented in the DBDD, the TLDD and the Software Development Folders (SDF). The detailed design will be presented at the Critical Design Review (CDR). This review will be held to assure that the requirements can be implemented and that the detail is sufficient to initiate coding. During this phase proto-flight code will be developed to add functional capabilities to process all CSCI external data, with the exception of some Fault Detection, Isolation, and Recovery (FDIR) and error handling, and to continue to substantiate sizing, timing, and interface loading estimates. Thorough CSU and CSC testing will not be performed in this phase. The proto-flight code will be an early version of the flight software.

The software development process concludes with the code/unit test/verification phase. In this phase, the proto-flight software developed thus far will be used as a baseline for development of the flight software, augmented with changes required as a result of the design process, and updates for standards compatibility. Proto-flight software used in the flight software will be upgraded during the code/unit test/verification phase and subjected to all tests defined in this plan for the flight software. Completion of the integration testing results in the release of the first version of the flight software, to be released via the Version Description Document (VDD). The flight software will then be available for formal qualification testing.

3.2.1.2 Flight Software Qualification

The CCS and NCS qualification process consists of a test requirements definition phase, a test procedure development phase, and a formal qualification test phase.

During the test requirements development phase, test requirements corresponding to the functional and performance requirements defined in the SRS will be developed and documented in the Software Test Plan (STP) for review at PDR.

During the test procedure development phase, qualification tests which show that the software meets its requirements will be developed and dry run. Test descriptions will be documented in the Software Test Description (STD) Volume I for review at CDR. Test procedures will be documented in the STD Volume II for review at Test Readiness Review (TRR).

The formal qualification test phase will use the developed procedures to test the flight software released at the conclusion of the flight software development process defined in section 3.2.1.1. At the completion of the formal qualification test phase, the CCS and NCS flight software is completely tested and available to support end item integration and test. The formal qualification test results will be documented in the Software Test Report (STR).

3.2.1.3 Simulation Software Development

The CES and NES simulation software development process is a phased development process consisting of requirements analysis, preliminary design, detailed design and code/unit test/verification phases.

The requirements analysis phase will produce an SRS. During this phase breadboard code will be developed to allow initial hardware/software integration, substantiate sizing and timing estimates, and provide functional capability for preliminary design phase integration. Thorough CSU and CSC testing will not be performed in this phase. Completion of the requirements analysis phase allows initiation of the preliminary design phase.

During the preliminary design phase the high level design of the software will be documented in a TLDD and a DBDD. During this phase brassboard code will be developed to add functional capabilities to input and output all CSCI external data and to substantiate sizing, timing, and interface loading estimates. Thorough CSU and CSC testing will not be performed in this phase.

The detailed design phase will follow the completion of the preliminary design phase. The detailed design will be documented in the DBDD and TLDD. During this phase proto-sim code will be developed to add functional capabilities to process all CSCI external data (with the exception of some FDIR and error handling) and to continue to substantiate sizing, timing, and interface loading estimates. Thorough CSU and CSC testing will not be performed in this phase. The proto-sim code will produce an early version of the simulation software.

The software development process concludes with the code/unit test/verification phase. In this phase, the proto-sim software developed thus far will be used as a baseline for development of the simulation software, augmented with changes required as a result of the design process, and updates for standards compatibility. Completion of the integration testing results in the release of the first version of the simulation software, to be released via the VDD. The simulation software will then be available to support formal qualification testing of the CCS and the NCS. The simulation software will also then be modified and released to support C&DH level testing in the SVF.

3.2.2 Activity Network

The sequential relationship of the various software development activities for each software release is shown in Figure 3.2.2-1, C&C IPT Activity Network.

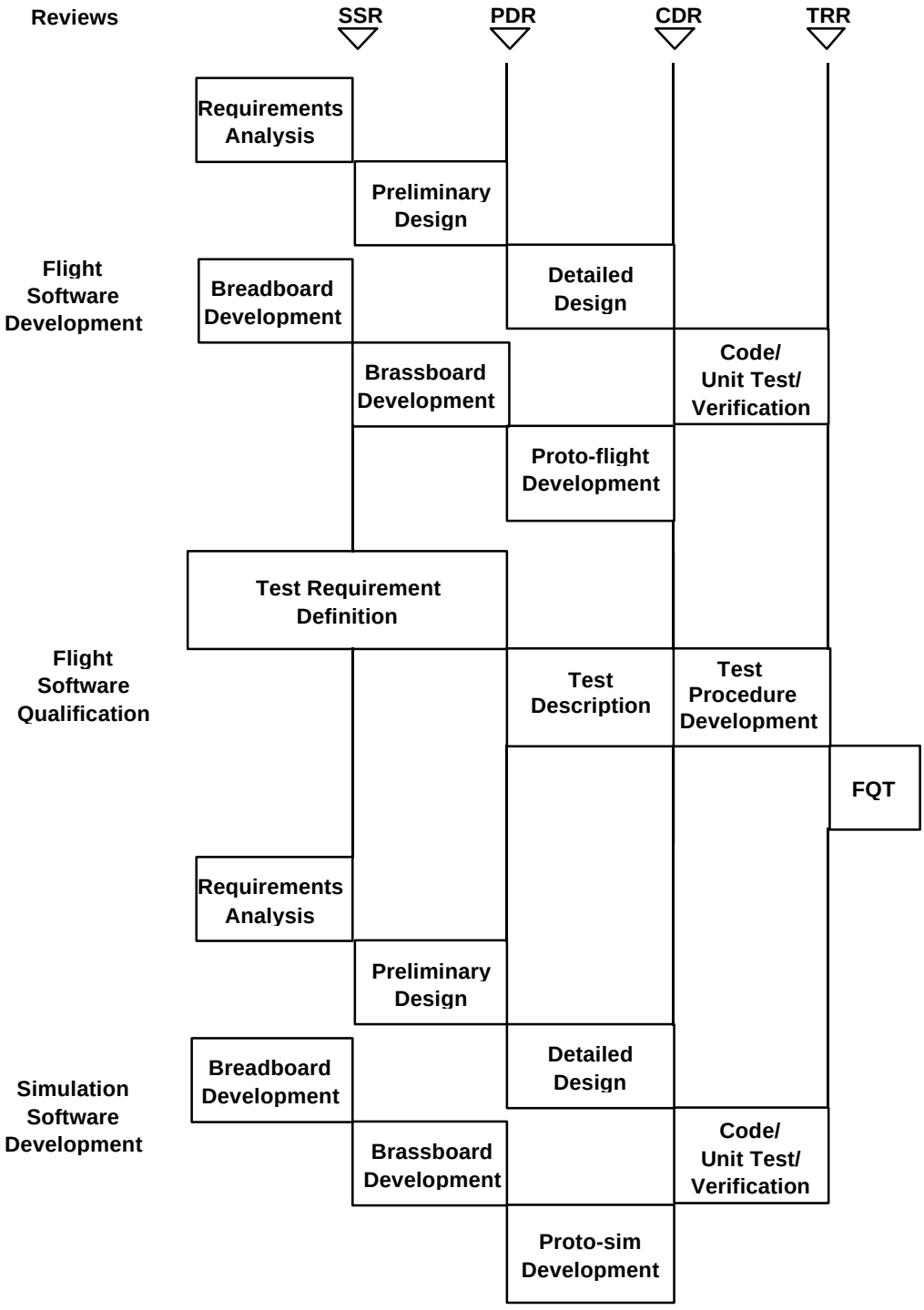


FIGURE 3.2.2-1 C&C IPT ACTIVITY NETWORK

3.2.3 Resource Identification

The resources required for the software development effort are described in section 4.1.

3.3 Risk Management

The C&C Software IPT will perform risk management for C&C Software IPT developed software as defined in the C&C IPT RMP, Volume 1. The C&C IPT RMP is compliant with D684-10017-1, Prime Contractor SDP, and D684-10054-1, Prime RMP. The C&C IPT RMP includes risk identification, assessment, analysis, abatement planning, and status monitoring as shown in Figure 3.3-1, C&C Risk Management Process.

C&C IPT RMP, Volume 2 documents the C&C risk management process. It includes risk lists, risk analysis reports, and risk abatement plans for each C&C software development cycle.

3.3.1 Risk Identification

The first step in risk management is identification of potential software risk areas. Each C&C Software IPT Group will identify their perceived risks based on symptoms identified by a questionnaire/checklist, metrics analysis, reviews, experience, and/or lessons learned. A standard risk identification questionnaire and checklist is provided in the C&C IPT RMP, Volume 1. The identified risks will be documented on a Risk Documentation Form. This form is also defined in C&C IPT RMP, Volume 1. The Risk Documentation Forms and subsequent assessments will be documented in C&C IPT RMP, Volume 2.

3.3.2 Risk Assessment

The characteristics of the risk will be quantified for analysis by the C&C Software IPT group who identified the potential risk. A Probability factor (Pf) and a Consequence factor (Cf) will be assigned for each risk identified. Pf is assessed by evaluating the maturity, complexity, dependency, and stability of the system under development. Cf is assessed by evaluating the technical, cost, and schedule aspects of the system under development. C&C IPT RMP, Volume 1 defines the guidelines for assigning these factors.

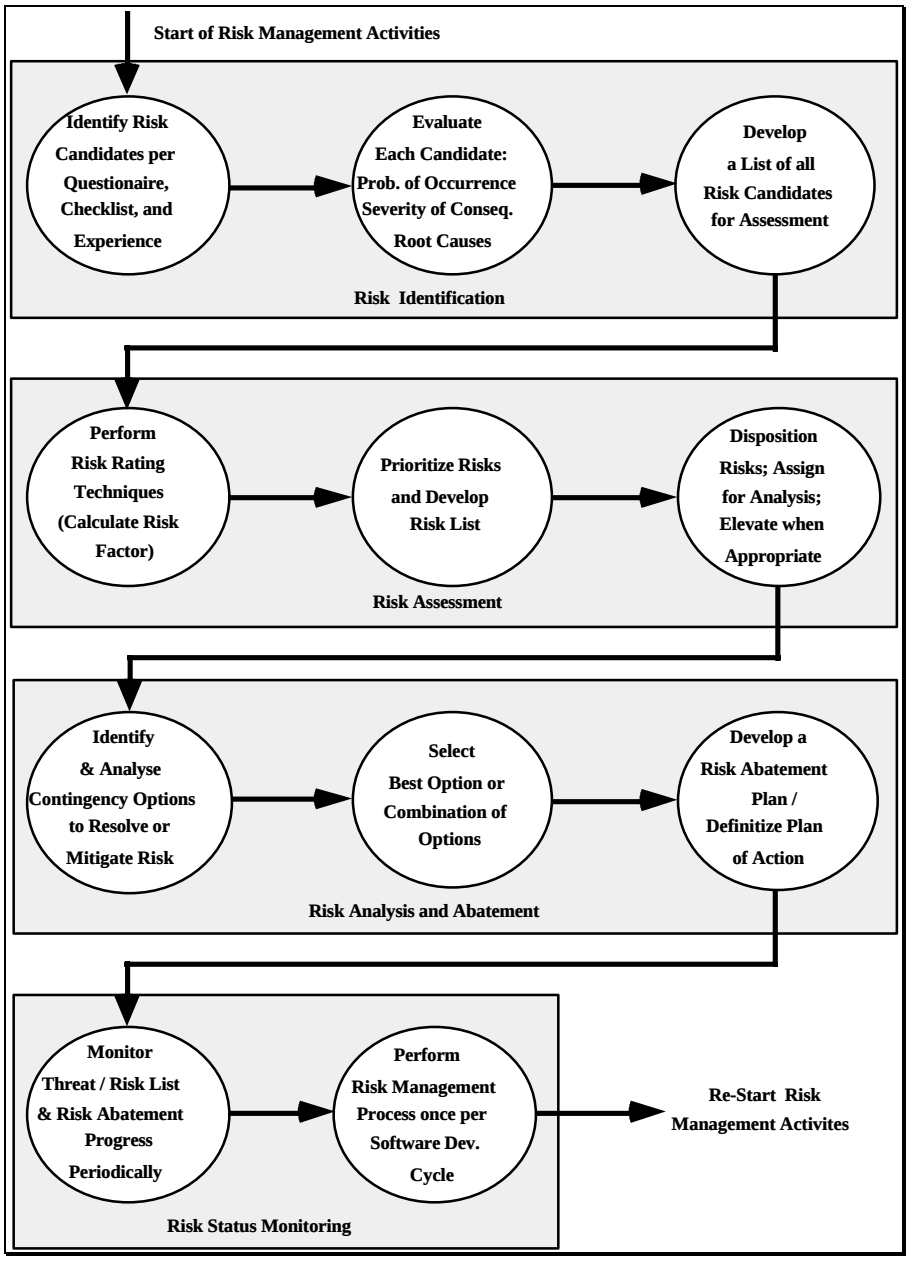


FIGURE 3.3-1 C&C RISK MANAGEMENT PROCESS

3.3.3 Risk Abatement

Risk abatement plans will be developed by the C&C Software IPT group or “ad hoc” working team assigned to work the risk. The risk abatement plan contains the following items:

a. A problem statement.

b. Potential impact and probability of occurrence,

c. Alternative corrective actions.

d. The recommended corrective action.

e. A recommended plan schedule.

f. Estimated program cost and schedule impacts.

This plan may present parallel development, work-around procedures, or backup capabilities that can be invoked to solve the problem. The risk abatement plans will be reviewed by the C&C Software IPT. The C&C Software IPT will approve appropriate risk aversion strategies. The status of software risk areas is reviewed at regular C&C Software IPT meetings and at regular internal program management reviews. The status, measured progress, and effectiveness of each abatement plan is reviewed. C&C Software IPT management then provides direction to continue, redirect resources, or close out the risk area upon successful completion of the abatement plan.

3.3.4 High Risk Areas

C&C Software IPT risks, associated risk abatement strategies, and results of the implemented strategy are documented in C&C IPT RMP, Volume 2. This document is updated as needed to reflect the current risk status for the C&C Software IPT.

3.4 Security

All Prime Contractor flight software development facilities will be consistent with sensitivity Criticality Level 1 as defined in NASA's Automated Information System (AIS) Security Manual. CCS and NCS software and data, once delivered to the Mission Build Operations Central Software Library, will be promoted to and maintained consistent with Criticality Level 3 when under the control of Mission Build Operations.

3.5 Interface with Associate Contractors

The Prime Contractor is responsible for the interface control process which is the basis for coordination of design and data management. The C&C Software IPTs will directly interface with other Product Groups, NASA, and International Partners. A Prime Interface Control Working Group (ICWG), chaired by a member of the Space Station AIT, integrates interface activities of the IPTs.

The C&C Software IPT will interface with the C&DH S/W Integ IPT during development of C&C Software IPT products/enditems. The C&C Software IPT will support the C&DH S/W Integ IPT and other C&DH organizations in the activities database definition and interface lashup. The C&C Software IPT will supply the Prime ICWG with the CCS and NCS interface data.

3.6 Interface with Software Independent Verification and Validation Agents

The C&C Software IPT will interface with the Space Station's Independent Verification and Validation (IV&V) agent per the Prime Contractor SDP.

3.7 Subcontractor Management

Subcontractors contributing to the development of the CCS and NCS are subcontracted and managed through various Prime contractor organizations (e.g., the C&DH S/W Integ IPT, the C&DH Subsystem AIT, and others). Subcontractor management is in accordance with section 3.7 of the Prime Contractor SDP.

3.8 Formal Reviews

Progress evaluation of CCS and NCS software development will be conducted as IPRs. These reviews are milestones that mark the transition of software products through the phases of the software development life cycle. (For more information regarding the software development life cycle used for C&C Flight Software development, see section 4.2.1.) IPRs of software products are conducted within the IPT structure as the products become available. The goal of an IPR is to bring together team representatives from various backgrounds, identify any issues with the product, and determine resolutions to these issues within the IPT environment. IPRs provide a status of the on-going product evaluation, obtain tentative approval for products scheduled to be baselined at the review, and identify any outstanding issues.

The IPRs for C&C Software IPT, and their applicability to flight software (CCS, NCS) and simulation software (CES, NES), are listed below in order of occurrence. The relationship between the review and the software development life cycle is shown in Figure 3.8-1, Reviews in the Software Development Life Cycle.

	Applicability	
	Flight S/W	Sim S/W
a. Software Specification Review (conducted by SMC) (SSR)	Yes	Yes
b. Preliminary Design Review	Yes	Yes
c. Critical Design Review	Yes	No
d. Test Readiness Review	Yes	Yes
e. Software Functional Configuration Audit (FCA)	Yes	No
f. Software Physical Configuration Audit (PCA)	Yes	No

The completion criteria for the reviews are:

a. To baseline the appropriate documentation pending the incorporation of approved issue resolutions

b. To obtain a directive from the C&DH Software Integration IPT to officially transition the products into the next phase of software development.

Participants in the formal reviews will include representatives from the other System AITs, Software Quality Assurance (SQA), SCM, IV&V, Software Test and Integration, S&MA, NASA Mission Operations Directorate, other NASA organizations, and the Prime Contractor.

The SSR is conducted by the SMC AIT. For the remaining reviews, the C&C Software IPT will coordinate review agendas and dates with the C&DH Software Integration IPT and provide a draft agenda and a draft set of briefing material to the C&DH Software Integration IPT a minimum of two weeks before the review date. The evaluation criteria for the review products will be defined in the corresponding review plans provided at least 1 month prior to each review.

The SSR, PDR, and CDR will primarily consist of an evaluation of documents to determine if the maturity is sufficient to proceed to the next development phase. For PDR and CDR the C&C Software IPT will provide copies of the review documentation to the C&DH Software Integration IPT a minimum of two weeks before the agreed upon review date.

Further detail on issue closure and tracking is documented in section 7. At the end of the review, the review participants will establish a date for re-submission of the products under review based on the volume of issues received and determine readiness to continue the activities supporting the next development phase for the CCS and the NCS.

3.8.1 Reviews Held for NASA (by the Prime Contractor)

The following reviews are held for NASA by the Prime Contractor. The C&C Software IPT will provide support as requested.

a. System Requirements Review

b. System Design Review (SDR)

3.8.2 Reviews Held for Prime Contractor Organizations

3.8.2.1 Software Specification Review

A review is conducted in the period following the release of the draft Software Requirements Specification. The SMC AIT will conduct the review in accordance with an SSR plan submitted for concurrence to the Software (SW) Integration AIT.

The purpose of the SSR is to 1) demonstrate accomplishment of the SSR intent as stated in the Prime Contractor SDP (D684-10017-1), using MIL-STD-1521B as a guideline and 2) establish customer agreement that the SSR milestone is complete and 3) establish agreement that the SRS provides a sound foundation to proceed with preliminary design.

3.8.2.2 Preliminary Design Review

A PDR is conducted by the C&C Software IPT during the period following the drafting of section 3 of the Top Level Design Document. The C&C IPT will conduct the review in accordance with an PDR plan submitted for concurrence to the SW Integration AIT.

The purpose of the PDR is to 1) demonstrate accomplishment of the PDR intent as stated in the Prime Contractor SDP (D684-10017-1), 2) establish customer agreement that the PDR milestone is complete and 3) establish agreement that the PDR provides a sound foundation to proceed with the detailed design phase.

3.8.2.3 Critical Design Review

A CDR is conducted by the C&C Software IPT. The C&C IPT will conduct the review in accordance with an CDR plan submitted for concurrence to the SW Integration AIT.

The purpose of the CDR is to 1) demonstrate accomplishment of the CDR intent as stated in the Prime Contractor SDP (D684-10017-1), 2) establish customer agreement that the CDR milestone is complete and 3) establish agreement that the detailed design of the software is correct, consistent and complete enough for development to continue to coding and informal testing, 4) provide a detailed basis for verifying design integrity and compatibility with CSCI requirements and assessment of formal test preparation.

3.8.2.4 Test Readiness Review

The TRR is the review of the flight software readiness to begin FQT of the CSCI. The review will be conducted by the C&C Software IPT after software test procedures are available and CSC integration is complete. The test procedures will have been executed in a "dry run" prior to their release for the review, and the procedures will be documented in Volume 2 of the STD. The C&C IPT will conduct the review in accordance with an TRR plan submitted for concurrence to the SW Integration AIT.

The TRR will be initiated when Volume 2 of the STD is available and FQT "dry run" is complete. The TRR will include presentations of any open issues, assigning actions to close the issues, and approval from the TRR board to start FQT. If the assigned actions require modifications to the test procedures, the flight software, or the test environment (including the simulations), the TRR board may provide conditional approval to begin software FQT, based on closing these actions.

The documents included in this review include Volume 2 of the STD, documentation of known flight software problems and possible work-arounds, and documentation of known problems with the test environment (including the simulations).

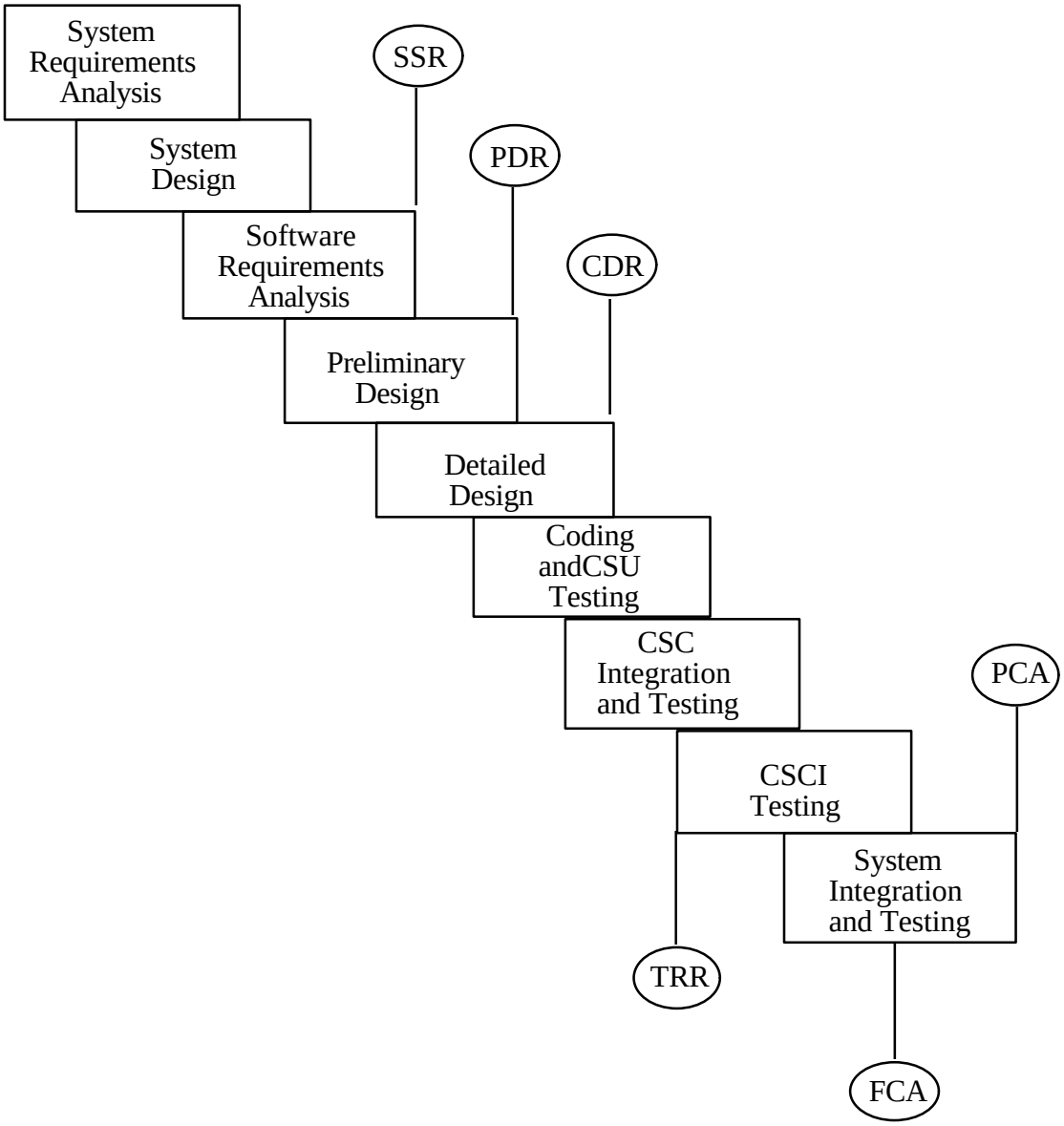


FIGURE 3.8-1 REVIEWS IN THE SOFTWARE DEVELOPMENT LIFE CYCLE

3.8.2.5 Functional Configuration Audit

The software FCA is conducted as a prerequisite to acceptance of the configuration item. The software FCA will be conducted by the C&C Software IPT after CSCI FQT.

The test data and results of the software FQT will be audited to determine whether the software fulfills its documented requirements and is ready for integration with the hardware for integrated subsystem-level FQT. The validity and completeness of the STR will be assessed.

The documents included in this review include any required updates to Volume 2 of the STD; the STR; and an updated SRS, if required.

3.8.2.6 Physical Configuration Audit

The software PCA is conducted as a prerequisite to establishing the product baseline. The software PCA will be conducted by the C&C Software IPT after the successful completion of the CSCI software FCA and FQT and prior to software acceptance.

The software PCA determines that the software which was tested and described by the test documentation accurately reflects the qualified, as-built, delivered CSCI. A detailed audit of the DBDD, TLDD and SDFs, which is comprised of the software design and source code listings, and the manuals for the flight software will be performed for format, traceability, consistency, and completeness. All previously baselined software documentation will be available for traceability analysis as required, and the VDD that describes the delivered CSCI will be available.

3.8.3 Stage Reviews

Stage software reviews will be held for each stage following the completion of C&C IPT reviews applicable to that stage. The stage reviews include all PG, IP, and ground interfaces applicable to that stage. All data/deliverables subsequent to the C&C IPT reviews must be presented at the stage reviews. Issues from the C&C IPT reviews that impact other developers and users must be brought to the stage reviews. Final closure of the C&C IPT reviews is contingent upon successful completion of the stage reviews.

3.9 Software Development Library (SDL)

The Software Development Library (SDL) is established to preserve and control software media and associated documents. The SDL is a repository for all controlled software products and documentation (with the exception of the SRS and Interface Control Document (ICD)), and will be maintained for the duration of the contract.

The SDL consists of three parts, the Engineering Design Library (EDL), the Integration Build Library (IBL), and the Configured Product Library (CPL). The EDL is controlled and managed at the Design Engineer level. The EDL is used as a working environment for software development, document development and software for CSU, CSC and pre-integration testing. The IBL is controlled and

managed at the C&C Software IPT level. The IBL is used as a build environment for integrated CSC and CSCI software, integration testing and developed documentation. The CPL is controlled and managed at the SCM IPT level. The CPL provides the released software for FQT testing and SVF testing and released documentation.

Figure 3.9-1, Software Development Library, depicts the SDL process. The Design Engineer has read only access to fetch software and documentation from the IBL and CPL. The read only access will ensure the integrity of the IBL and CPL. When the development of a software product in the EDL is complete, the software product is submitted to the IBL via a Mail message.

Following verification of the Mail message, the IBL is updated. After the software in the IBL has completed integration testing, the software is submitted to the CPL via a Request To Update (RTU). Following a successful audit of the RTU, the CPL is updated. Once in the CPL, an authorized Software Problem Report (SPR) is required to change any software or documentation.

The integrity of the IBL and the CPL is maintained in the following ways:

a. The Design Engineer has Read/Execute access to the IBL and CPL.

b. The C&C IPT has Read/Write/Execute/Delete access in the IBL and Read/Execute access to the CPL.

c. The SCM IPT has Read/Write/Execute/Delete access to the CPL.

d. After data or documentation are submitted to the IBL or CPL, no change can be made directly to that data.

e. Changed data submitted for incorporation into the CPL will be authorized by an SPR after the initial release of the product.

f. Records of controlled software item configurations, SPRs and change tracking will be maintained in the CPL.

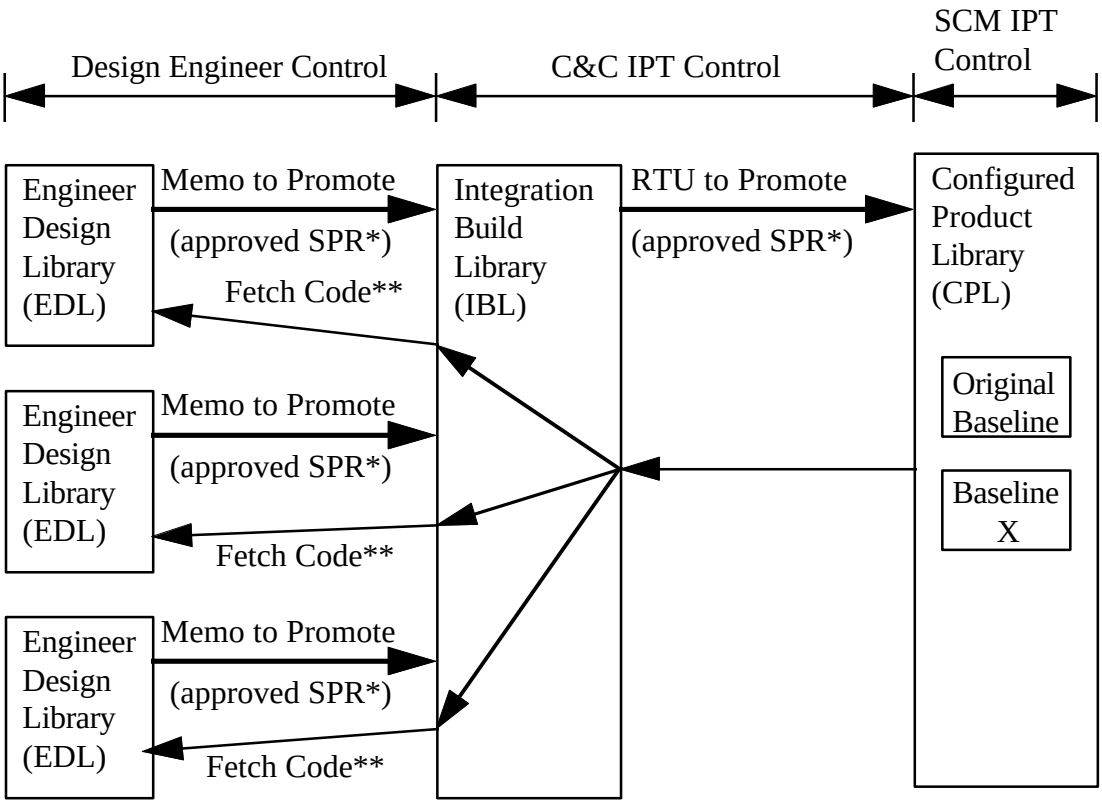
The process for submitting products to the CPL and for making revisions to CPL controlled products is described in section 7.3.1.

3.10 Corrective Action Process

The process used to identify, analyze, correct and track problems found in a controlled software product is described in section 7.3.1.

3.11 Problem/Change report

The problem change report is described in section 7.3.2.



NOTE: * Only after the initial release
** If the software/documentation not in IBL, fetch from CPL

FIGURE 3.9-1 SOFTWARE DEVELOPMENT LIBRARY

3.12 Software Metrics Management

3.12.1 Organization and Resources

The C&C Software IPT manager is responsible for reporting software metrics to the Software Integration IPT.

3.12.2 Purpose and Scope

The goals of software metrics are to provide management visibility into the software development process and to promote the timely development of quality software products. The analysis and assessment of software metrics provides early warning signals to highlight potential development, test, integration, or schedule problems before they become unrecoverable, causing schedule disconnects and/or slippage.

Software metrics reporting applies to the CCS, NCS, CES, and NES software for the C&C Software IPT development activities.

3.12.3 Software Metric Reporting Methodology

C&C Software IPT metrics are reported at the CSCI level. Software metrics include plans, estimates, and actual results for the reporting period.

3.12.4 Software Metrics

Software metrics, or software management indicators, as defined in paragraph 3.12.4 of the Prime SDP are reported periodically (quarterly) to the Software Integration IPT.

The C&C Software IPT will periodically measure and report items such as Central Processing Unit (CPU), memory, bus bandwidth utilization, and Source Lines of Code (SLOC). In addition the C&C Software IPT will periodically measure and report the planned versus actual number of CSCs designed, coded and tested and the planned versus actual number of formal test procedures developed and performed.

The following software metrics are maintained internal to the C&C Software IPT:

a. SPR resolution and processing time
b. Defect characterization and profile
c. Requirements stability and resolution
d. Cost/scheduling deviations
e. Staffing profile

3.13 Flight Software Builds

The C&C Software IPT is responsible for releasing the CCS and NCS CSCIs to the Software Support Library of the MBF. The components that comprise the CSCI are specified in the respective VDD. The VDD also specifies the instructions needed to perform the CSCI build utilizing the software configuration management and build tools of the MBF.

3.14 Firmware Management

Not applicable.

3.15 Shared MDM Integration Strategy

The C&C Software IPT is the "CSCI Owner" for the CCS which will reside in the C&C MDMs and the NCS which will reside in the Node 1 MDM. As such the C&C Software IPT will adhere to the responsibilities as outlined in section 3.15.1 item B. of the Prime Contractor SDP.

The C&C Software IPT is not an "MDM Owner" (see section 3.15.1 item A. of the Prime Contractor SDP) nor is the C&C Software IPT a "CSC Supplier" (see section 3.15.1 item C. of the Prime Contractor SDP).

With the exceptions of MDM Services (see section 4.5) and Timeliner (which is GFE), no software CSCs are supplied for incorporation into CCS or NCS.

4. SOFTWARE ENGINEERING

4.1 Organization and Resources – Software Engineering

This section describes the organization and resources to be used in developing the flight software and related support software.

4.1.1 Organizational Structure – Software Engineering

The software engineering organization for the C&C Software IPT is represented as the C&C Flight Software Development, C&C Simulation Software Development and C&C Systems Engineering Groups in Figure 3.1.3-1, C&C Software IPT Organization. The software engineering responsibilities are identified in section 3.1.3.

The details of the SMC AIT organization are described in the SMC TEP.

4.1.2 Personnel – Software Engineering

The personnel assigned to software engineering are consistent with the personnel requirements identified in section 3.1.4. Personnel requirements, skills, and staffing levels are maintained by the C&C Software IPT and SMC AIT managers in accordance with Prime Contractor requirements. Personnel skills and title information are described in the C&C IPT and SMC TEPs.

4.1.3 Software Engineering Environment

The SEE is used to specify, develop and manage flight software. A subset of the SEE has been mandated by the Prime Contractor for program wide use to build MDM based software deliverables. The CCS and NCS CSCIs are an MDM-based software deliverable. The mandated SEE products are listed in the Prime Contractor SDP section 4.1.3.

The SEE subset used for C&C Flight Software and simulation development is located in the C&C PSPF. The C&C PSPF is one segment of the SDIL.

The C&C PSPF contains Contractor resources and GFE. Sections 4.1.3.1 and 4.1.3.2 describe C&C PSPF SEE resources in terms of software, hardware, and firmware items.

4.1.3.1 Software Items

4.1.3.1.1 General Purpose Software

The C&C PSPF uses general purpose software tools for managing the C&C software development process. These GFE tools consist of the following:

a. Microsoft Project – planning and management

b. Deneba Systems Canvas – graphical editor

c. Microsoft Word – word processor

d. Microsoft Excel – spreadsheet

e. Microsoft PowerPoint – presentation materials editor

f. Fastrack - scheduler

4.1.3.1.2 Systems Software

Systems software tools are used by the C&C IPT to support software development. These tools assist in requirements collection and analysis, flight software and simulation design, flight software development, simulation development, test script development, configuration management, and documentation. At this time the following systems software tools are planned to be used to develop C&C flight software and simulations:

a. Digital Equipment Corporation (DEC) Virtual Address Extension (VAX)/Virtual Memory System (VMS) Operating System – general purpose operating system for development and general operations such as backup/restore, file transfer, and user account authentication

b. DEC VAX/VMS Language Sensitive Editor – general purpose text editor used to write C&C Flight Software and simulations

c. DEC VAX/VMS Code Management System (CMS) – general purpose configuration management tool

d. DEC VAX/VMS Module Management System – general purpose configuration management support tool

e. DEC VAX/VMS DOD Network Protocols – File Transfer Protocol (FTP), Telnet, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Protocol (IP). Supports data transfers between DEC and non-DEC host computers

f. DEC VAX/VMS DECNet Network Protocols – DEC proprietary protocols for networking VAX hosts

g. Alsys Ada Compiler – DEC VAX/VMS based cross-compiler for the C&C MDM Intel 80386 based target processor

h. PharLap Assembler – DEC VAX/VMS based cross-assembler for the C&C MDM Intel 80386 based target processor

i. PharLap Linker – DEC VAX/VMS based link editor for linking Alsys and PharLap assembler generated object modules into an executable C&C image

j. Integrated Systems Incorporated (ISI) Technologies Advanced Sim Package – real-time process control simulation and code development tool

k. ISI Technologies AutoCode Ada – companion auto-code generation tool for the ISI simulation tools

l. ISI Technologies DocumentIt – DOD-STD-2167A compatible documentation support tool for the ISI simulation tools

m. MATRIXx – MDM simulation development and test tool which runs on the MDM Applications Test Environment (MATE)

n. To Be Determined (TBD) Database – for collecting and maintaining ICD information required to implement C&C Flight Software and simulations

o. ICONIX Power Tools, from ICONIX Software Engineering, Inc. - requirements, design, and documentation support tools

4.1.3.2 Hardware and Firmware Items

The C&C Software IPT develops and operates the C&C PSPF required to accomplish software development for its software deliverables. The C&C PSPF hardware configuration is shown in Figure 4.1.3.2-1, C&C Development Hardware Architecture. The hardware and firmware items include the following:

a. Four Enhanced MDM functionally equivalent units (FEUs) – two to support CCS development and two to support CCS FQT.

b. Five MATE 3s – two to support CCS development, one to support CCS FQT, one to support NCS development, one to support NCS FQT, and each containing the following features:

1. An Intel 80486-based CPU
2. Two MIL-STD-1553 bus cards
3. 300-megabyte disk
4. A Small-Computer System Interface (SCSI) card (for input/output simulation access)
5. An Ethernet card

c. Digital Equipment Corporation (DEC) Virtual Address Extension (VAX) 10620 (host computer) – for software code development and configuration management

d. Twelve dedicated DEC VAX 4000 workstations – to host the MDM cross-compiler, debuggers, editors, CMS, and the ISI tool set; additional workstations will be shared with the SVF

e. An integrated local area network of Macintosh and Personal Computer (PC)-compatible workstations – to host the general-purpose software tools for software planning, management, and design

f. Four standard MDM FEUs – two to support NCS development and two to support NCS FQT.

g. A Tektronix 640A oscilloscope

h. A DAS 9200 logic analyzer

i. An Ancot 201 SCSI bus Analyzer

j. One IBM PS2/95 – acting as a personal computer system simulation host

k. Two IBM PS2/80 – for downloading MDM firmware

l. Two Microtec 80386 in-circuit emulators

m. Six MIL-STD-1553 bus patch panels

n. Miscellaneous items such as printers, secondary storage (e.g., tapes), four floating monitors, three 3.5-inch floppy drives, and associated test and instrumentation equipment

o. Four SUN SPARC 5 and one SUN SPARC 20 - one of each of the five (three CCS and two NCS) development/test stations

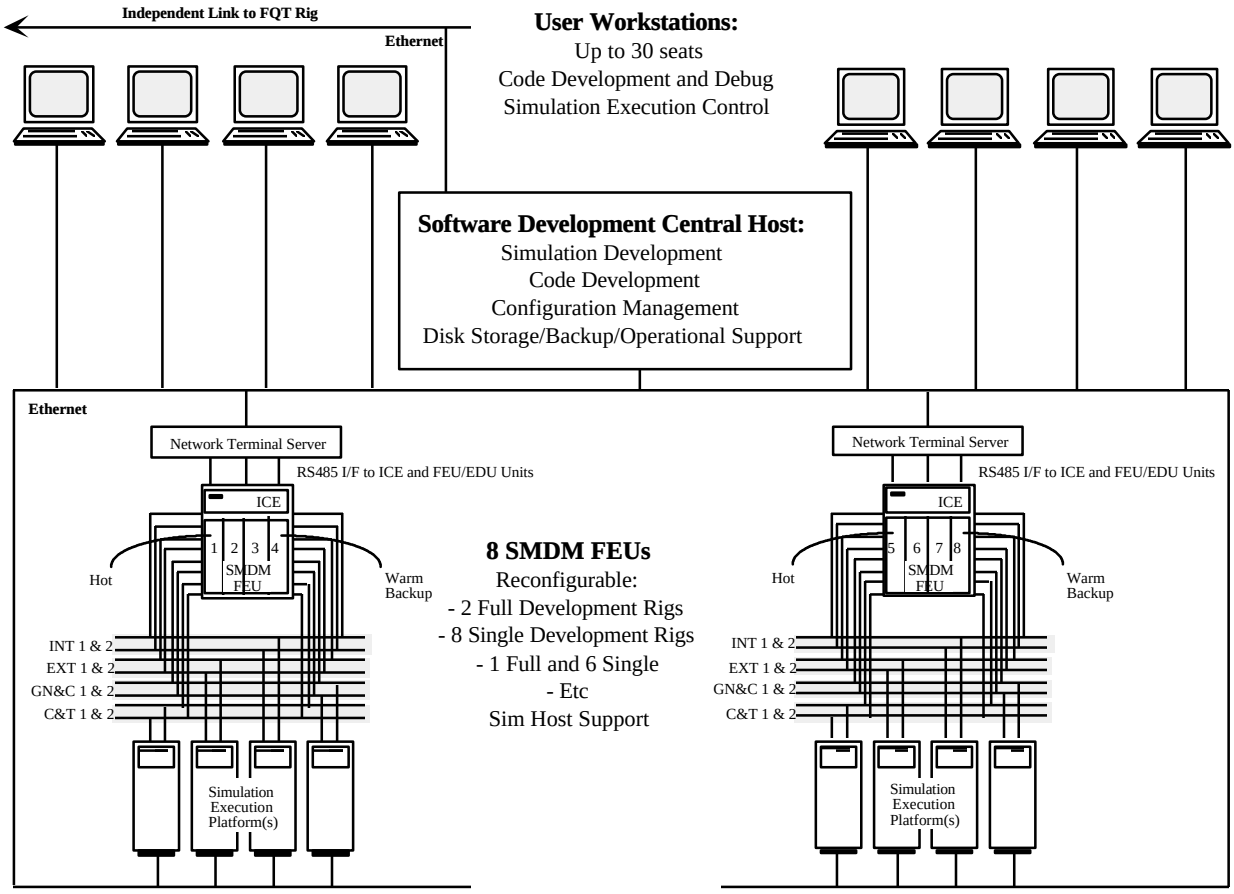


FIGURE 4.1.3.2-1 C&C DEVELOPMENT HARDWARE ARCHITECTURE

4.1.3.3 Proprietary Nature and Government Rights

Hardware and software purchased from commercial vendors and provided to development organizations or program facilities are used according to appropriate restricted access rights established by the particular licensing agreement.

Software and firmware identified in the Prime Software SDP as newly developed will be furnished to the government in accordance with the Rights in Data-General Clause of Federal Acquisition Regulation (FAR) 52.227-14, together with NFS 18.52.227-14.

4.1.3.4 Installation, Control, and Maintenance

The SDIL IPT will maintain configuration control of the facility in accordance with SDIL-0003, SDIL Configuration Management Handbook. The SDIL CM Handbook identifies the processes and responsibilities for the control of facility systems software and equipment. Further detail is provided in the Maintenance & Operations (M&O) Standard Operating Procedures. The SDIL IPT will be responsible for the installation, retention, and cataloging of SEE products (MDMs, MATEs, etc.) provided for use in the C&C PSPF. The installation of SEE will be performed according to the procedures provided by the M&O IPT, as a sub-team of the SDIL IPT, and in accordance with the specific instruction supplied with the SEE. General purpose software, as identified in paragraph 4.1.3.1.1, and the associated PC and Macintosh workstations are the responsibility of NASA Information Systems.

4.2 Software Standards and Procedures

CCS, NCS, CES and NES software development will comply with the standards and procedures contained in the C&C Software SSPS. These standards and procedures are based on DOD-STD-2167A, the Prime Contractor SDP, and the Prime Contractor SSPS.

Exceptions to the Prime Contractor SSPS and DOD-STD-2167A are noted in section 9.2.

4.2.1 Software Development Techniques and Methodologies

The CCS and NCS CSCIs will be developed in the phased engineering approach shown in Figure 4.2.1-1, Flight Software Development Phases, including software requirements analysis, preliminary design, detailed design, coding, unit testing, integration testing, and CSCI testing. Also shown in the figure are the documents produced as the result of the development activity.

The software requirements, design, and code will be functionally grouped into Top Level Computer Software Components (TLCSCs), CSCs, and CSUs as shown in the example in Figure 4.2.1-2, System Breakdown and CSCI Decomposition. The following sections describe the tasks and event flows associated with each phase.

4.2.1.1 Software Requirements Analysis

The primary software product during this phase is a draft SRS. The SMC AIT is responsible for the SRS and is the primary team for the development of the requirements through performance of system and requirement allocation analysis as well as analysis of system architectures and designs.

During the software requirements analysis phase, the software designers will support system analysis, development of ICD Part 1's, requirements allocation, and design trade studies. Supportive of this task will be the development of a top-level architecture which includes identification of major software functions, creation of function hierarchy diagrams and functional/data flow diagrams, specification of hardware/software interfaces, and sizing/timing estimates.

In parallel with the requirements analysis phase is the development of breadboard software. This development of selected software functions will provide initial hardware/software integration, substantiate sizing and timing estimates, and provide the functional capability to support the preliminary design phase brassboard development.

4.2.1.2 Preliminary Design

During the Preliminary Design phase the software requirements and top-level architecture for the software will be further developed. The SRS will be baselined and a draft software test plan will be produced. The preliminary design phase activities consist of the following:

a. The CSCI's preliminary design will be documented in a TLDD, DBDD, Software User's Manual (SUM), and Software Development Folders (SDFs). Preliminary design documentation standards are described in section 4.2.3.

b. Hardware/software interface will be defined.

c. Additional design analyses, rationale, and results of trade studies that may be required to understand the design will be performed.

d. The formal qualification tests to be conducted to comply with the SRS requirements will be defined. These qualification requirements will be documented in a STP.

Development of the brassboard software will take place during the preliminary design phase. Functional capabilities to support the CSCI external interfaces will be provided. These functional capabilities support further design substantiation and provide an early test bed for test procedure development.

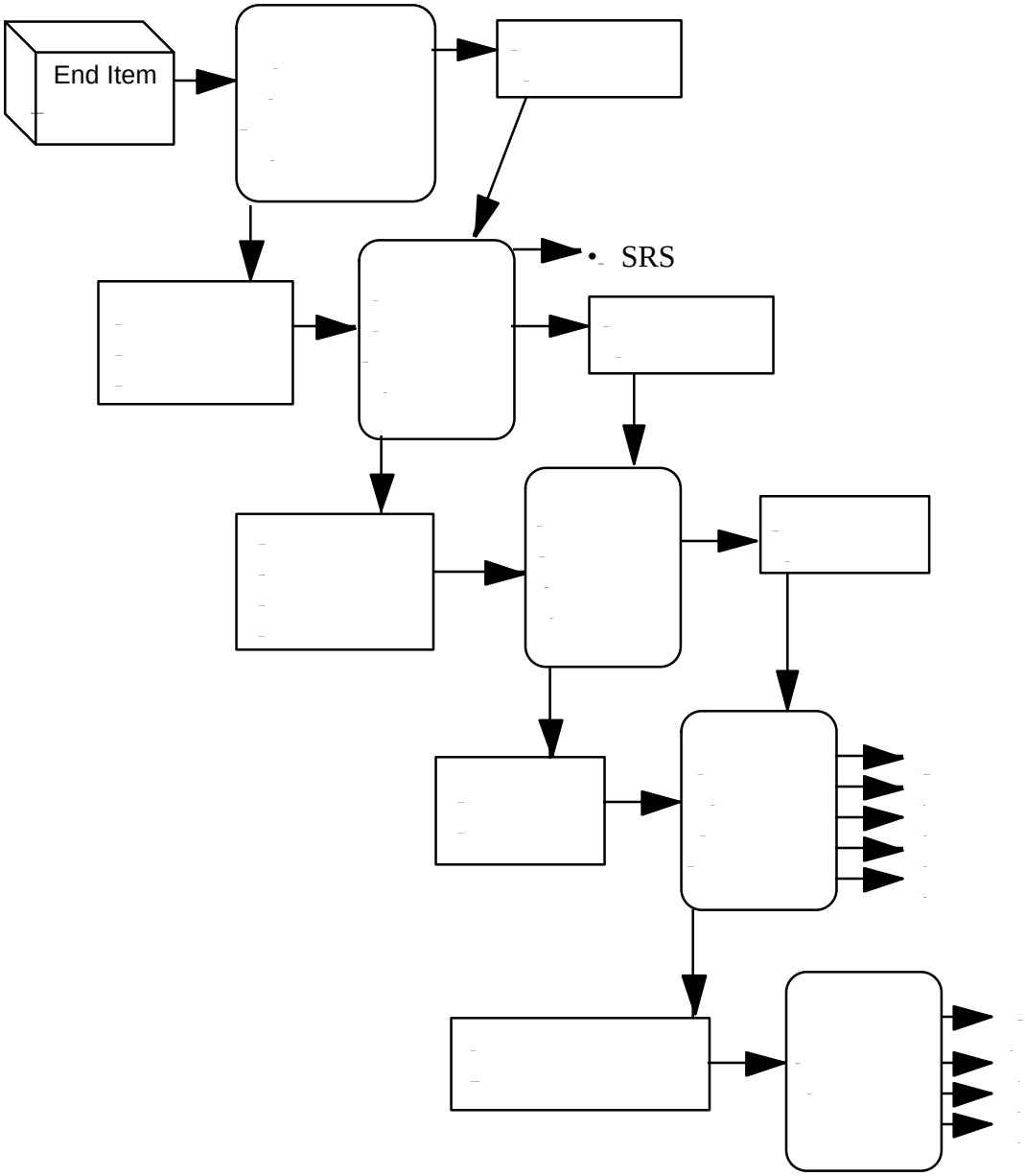


Figure 4.2.1-1 Flight Software Development Phases

FIGURE 4.2.1-1 FLIGHT SOFTWARE DEVELOPMENT PHASES

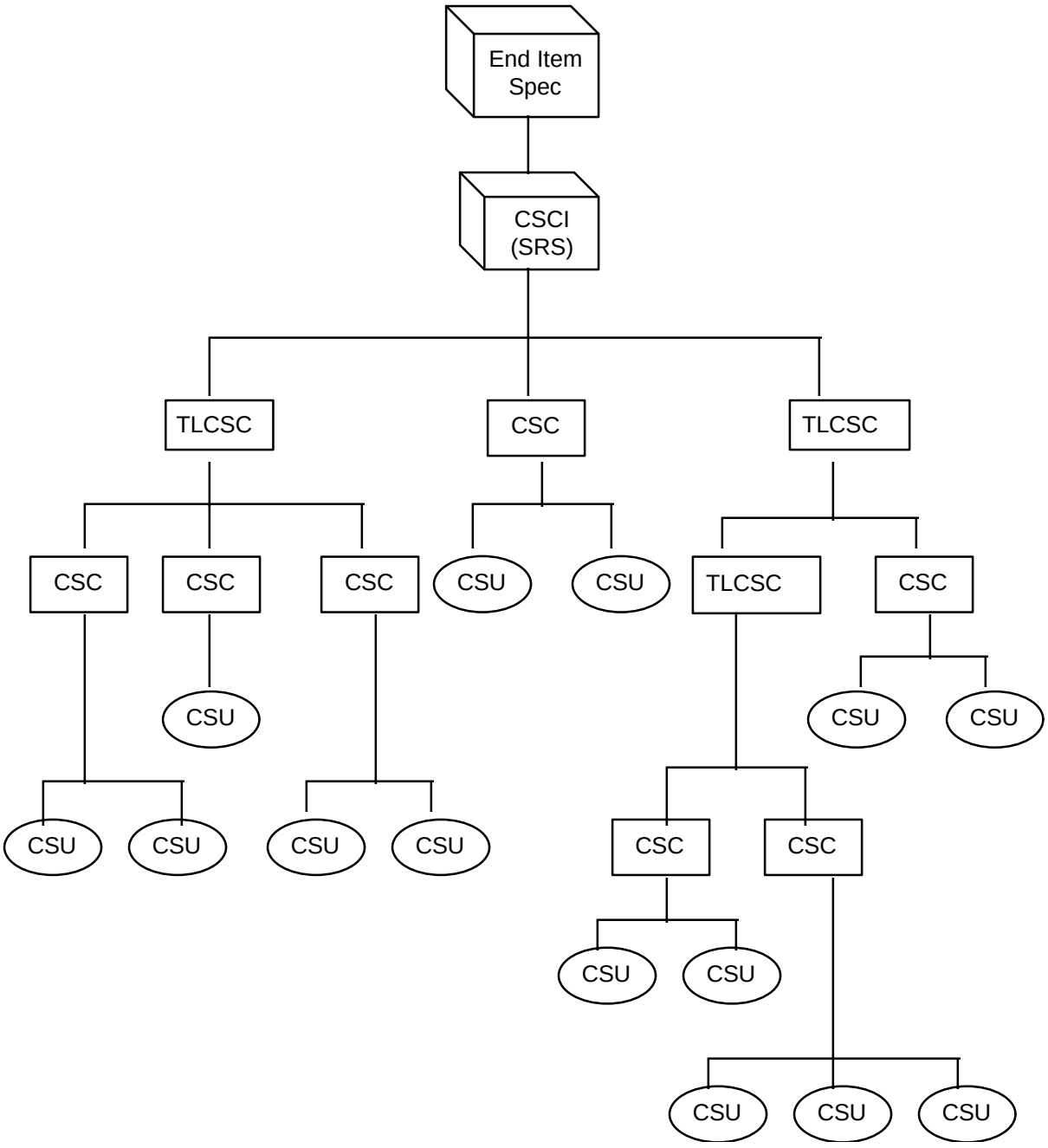


FIGURE 4.2.1-2 SYSTEM BREAKDOWN AND CSCI DECOMPOSITION

4.2.1.3 Detailed Design

During the Detailed Design phase the software's design will be completed. The detailed design phase activities consist of the following:

a. The requirements from each CSC will be allocated to the CSUs and the design of each CSU will be established. Each CSC and its subordinate CSUs will be documented in the SDFs and the Ada Package Specifications.

b. Additional design analyses, rationale, and results of trade studies that may be required to understand the design will be produced.

c. Test responsibilities, test cases, and test schedules for CSC integration and testing will be established.

d. The test cases for the formal qualification tests specified in the STP will be identified and described.

Proto-flight software will be developed adding operational capability. This operational capability represents the full operational capability of the CSCI minus some FDIR capabilities and some error handling. Results of the proto-flight development activity will in turn be included in the detailed design and documentation.

4.2.1.4 Coding and CSU Testing

The Coding and CSU Testing phase activities consist of the following:

a. Test procedures for conducting each CSU test will be developed. These CSU test procedures will be recorded in the corresponding CSU SDF.

b. Each CSU will be coded and tested. CSU testing will, as a minimum, ensure that all logic paths in a CSU are exercised. Note that code and/or test cases may already exist in the proto-flight software. Use of this code will be encouraged due to design maturity, as long as the design requirements and coding standards are met, and the code passes the CSU test.

c. All necessary revisions to the design documentation and code will be made, all necessary retesting will be performed, and the SDFs for all CSUs that undergo design or coding changes based upon CSU tests will be updated.

d. The test procedures for conducting CSC integration and testing will be developed. These procedures will be recorded in the CSC SDFs.

The coding and naming standards are contained in the C&C SSPS. The detailed processes and standards for performing CSU/CSC testing and configuration control of the code are also contained in the C&C SSPS.

4.2.1.5 CSC Integration and Testing

The CSC Integration and Testing phase activities consist of the following:

a. CSC integration and testing will be performed to ensure that the algorithms and logic employed by each CSC are correct, that the CSC satisfies its allocated SRS capabilities, and that all interfaces between CSUs are exercised.

b. The results of CSC integration and testing will be recorded in the corresponding SDFs.

c. All necessary revisions to the design documentation and code will be made, all necessary retesting will be performed, and the SDFs for all CSUs that undergo design or coding changes based upon results of all tests performed will be updated.

d. The procedures for the set-up, formal testing, and analysis of formal test results for the qualification test cases will be developed.

e. CSC to CSC integration and test will be performed to ensure that the algorithms and logic performed by multiple CSCs are correct and that all interfaces between CSCs are exercised.

f. CSCI testing will be performed to ensure that the procedures for the FQT are complete and accurate and that the CSCI is ready for FQT.

The detailed processes and standards for performing informal CSC to CSC integration and CSCI level testing are contained in the C&C SSPS.

4.2.1.6 CSCI Testing

The CSCI Testing phase will include Formal Qualification Testing of the software, as described in section 5.0. CSCI testing activities consist of the following:

a. The FQT activities will be performed in accordance with the procedures documented in the software test procedures in the STD Volume 2.

b. The results of the FQT will be recorded in a STR, including an as-run set of procedures.

c. All necessary revisions to the design documentation and code will be made, all necessary retesting will be performed, and the SDFs for all CSUs that undergo design or coding changes based upon results of FQT will be updated.

The detailed processes and standards for performing CSCI FQT are contained in the C&C SSPS.

4.2.2 Software Development Folders

The SDF is a structured collection of documents, data, and information for each flight software CSCI and each CSC within the flight software CSCIs. Including both paper (files) and electronic media, the SDF provides an orderly record of the development history of each CSCI/CSC. A header page included with each CSCI's/CSCs SDF will identify the location of all electronic data pertinent to the CSCI/CSC.

Included in the SDFs will be requirements allocations, design documentation, source code, test cases, and test results. Details of the format and contents of the SDFs are contained in the C&C Software SSPS.

Each SDF will be the responsibility of the software design engineers assigned to that piece of software. The software lead will conduct periodic reviews of the SDF to ensure completeness. SDFs will be maintained and controlled by the C&C Software IPT, and will be available for customer review.

4.2.3 Design Standards

The design standards will be defined in the C&C Software SSPS. These design standards will comply with standards and procedures contained in the Prime Contractor SSPS with the exceptions noted in section 9.2. The design standards will be used for the development of the flight software and the simulation software.

4.2.4 Coding Standards

The coding standards will be defined in the C&C Software SSPS. These coding standards will comply with standards and procedures contained in the Prime Contractor SSPS except as noted in section 9.2 of this plan. The coding standards will be used for the development of the flight software and the simulation software.

4.3 Non-developmental Software

Non-developmental software, as it applies to the C&C Software IPT software development, is GFE. GFE used in the CCS is identified in section 3.1.2 of this plan. The CCS and NCS use the Honeywell Utilities and the Alsys Ada Run Time Environment.

4.4 Non-flight Software

4.4.1 Non-flight Software Development Techniques and Methodologies

The CES and NES CSCIs will be developed in the phased engineering approach shown in Figure 4.4.1-1, Non-flight Software Development Phases, including software requirements analysis, preliminary design, detailed design, coding, unit testing, and integration testing. Also shown in the figure are the documents produced as the result of the development activity.

The software requirements, design, and code will be functionally grouped into TLCSCs, CSCs, and CSUs as shown in the example in Figure 4.4.1-2, System Breakdown and CSCI Decomposition. The following sections describe the tasks and event flows associated with each phase.

4.4.1.1 Software Requirements Analysis

During the software requirements analysis phase, the software designers will perform system analysis, requirements allocation, and design trade studies. The primary software product during this phase is a draft SRS. Supportive of this task will be the development of a top-level architecture which includes identification of major software functions, creation of function hierarchy diagrams and functional/data flow diagrams, specification of hardware/software interfaces, and sizing/timing estimates.

In parallel with the requirements analysis phase is the development of breadboard software. This development of selected software functions will provide initial hardware/software integration, substantiate sizing and timing estimates, and provide the functional capability to support the preliminary design phase brassboard development.

4.4.1.2 Preliminary Design

During the Preliminary Design phase the software requirements and top-level architecture for the software will be further developed. The preliminary design phase activities consist of the following:

a. The CSCI's preliminary design will be documented in a TLDD, DBDD, and SDFs. Preliminary design documentation standards are described in section 4.2.3.

b. Hardware/software interface will be defined.

c. Additional design analyses, rationale, and results of trade studies that may be required to understand the design will be performed.

d. Requirements for conducting informal CSC integration and testing will be developed. These informal CSC integration and testing requirements will stress the software at the limits of its specified requirements.

Development of brassboard software will take place during the preliminary design phase. Functional capabilities to support the CSCI external interfaces will be provided. These functional capabilities support further design substantiation and provide an early test bed for flight software test procedure development.

4.4.1.3 Detailed Design

During the Detailed Design phase the software's design will be completed. The detailed design phase activities consist of the following:

a. The requirements from each CSC will be allocated to the CSUs and the design of each CSU will be established. Each CSC and its subordinate CSUs will be documented in the SDFs and the Ada Package Specifications.

b. Additional design analyses, rationale, and results of trade studies that may be required to understand the design will be produced.

c. Test responsibilities, test cases, and test schedules for CSC integration and testing will be established.

Proto-sim software will be developed adding operational capability. This operational capability represents the full operational capability of the CSCI minus some error handling capabilities. Results of the proto-sim activity will in turn be included in the detailed design and documentation.

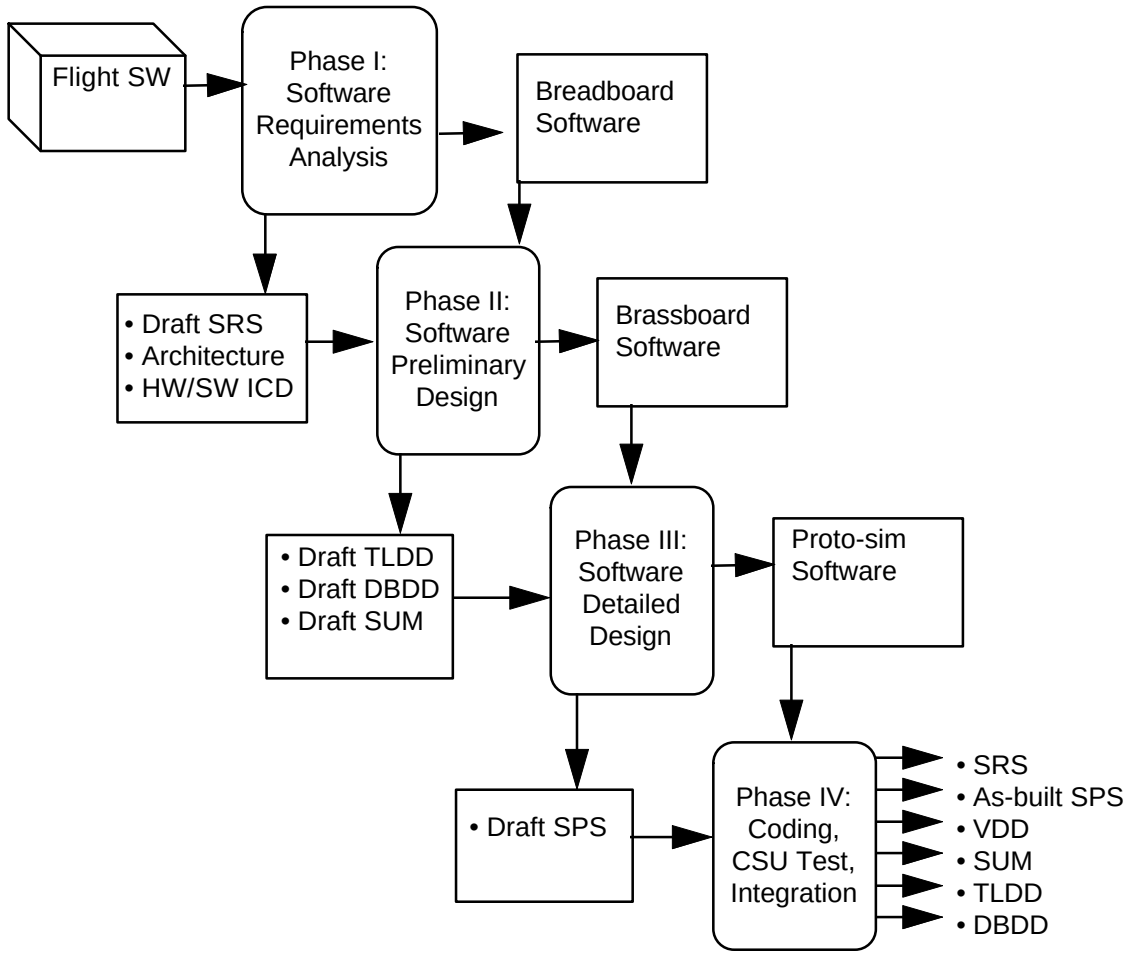


Figure 4.4.1-1 Non-flight Software Development Phases

FIGURE 4.4.1-1 NON-FLIGHT SOFTWARE DEVELOPMENT PHASES

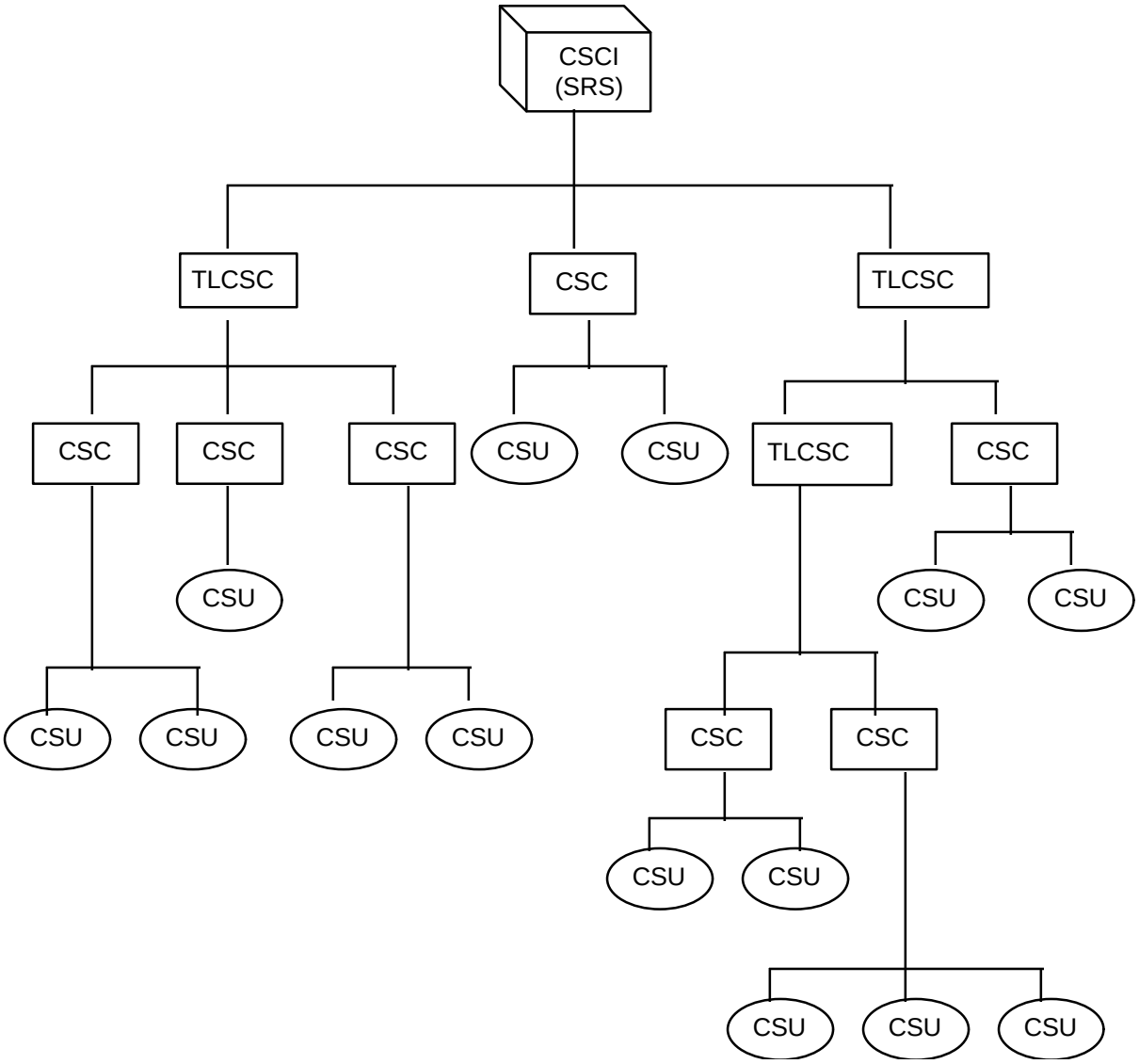


FIGURE 4.4.1-2 SYSTEM BREAKDOWN AND CSCI DECOMPOSITION

4.4.1.4 Coding and CSU Testing

The Coding and CSU Testing phase activities consist of the following:

a. Test procedures for conducting each CSU test will be developed. These CSU test procedures will be recorded in the corresponding CSU SDF.

b. Each CSU will be coded and tested. CSU testing will, as a minimum, ensure that all logic paths in a CSU are exercised. Note that code and/or test cases may already exist in the proto-sim software. Use of this code will be encouraged due to design maturity, as long as the design requirements and coding standards are met, and the code passes the CSU test.

c. All necessary revisions to the design documentation and code will be made, all necessary retesting will be performed, and the SDFs for all CSUs that undergo design or coding changes based upon CSU tests will be updated.

d. The test procedures for conducting CSC integration and testing will be developed. These procedures will be recorded in the CSC SDFs.

The coding and naming standards are contained in the C&C SSPS. The detailed processes and standards for performing CSU/CSC testing and configuration control of the code are also contained in the C&C SSPS.

4.4.1.5 CSC Integration and Testing

The CSC Integration and Testing phase activities consist of the following:

a. CSC integration and testing will be performed to ensure that the algorithms and logic employed by each CSC is correct, that the CSC satisfies its allocated SRS capabilities, and that all interfaces between CSUs are exercised.

b. The results of CSC integration and testing will be recorded in the corresponding SDFs.

c. All necessary revisions to the design documentation and code will be made, all necessary retesting will be performed, and the SDFs for all CSUs that undergo design or coding changes based upon results of all tests performed will be updated.

d. CSC to CSC integration and test will be performed to ensure that the algorithms and logic performed by multiple CSCs are correct and that all interfaces between CSCs are exercised.

e. CSCI testing will be performed to ensure that the CSCI is ready to support flight software FQT.

The detailed processes and standards for performing informal CSC to CSC integration and CSCI level testing are contained in the C&C SSPS.

4.5 MDM Services Software

MDM Services Software is the Alsys Ada Run-Time Environment (RTE) and the software utilities developed by the MDM provider to support MDM initialization, MDM health monitoring, low level 1553

communication and list based sensor and effector input/output. The portions of the Alsys Ada RTE and the software utilities that are used and how they relate to the CCS and the NCS will be documented as part of the CCS and the NCS design and as-built documentation. The software utilities will be extensively tested in the CSU and CSC level tests for those CSUs and CSCs that utilize these software utilities. These software utilities will also be implicitly tested (due to the nature of the functions provided by these software utilities) extensively during CSC integration testing and FQT. This testing will be performed to ensure that MDM Services Software used in CCS and NCS together with the C&C Software IPT developed CCS and NCS software satisfy the requirements of the CCS and NCS CSCIs.

5. FORMAL QUALIFICATION TESTING

5.1 Organization and resources

5.1.1 Organizational structure – formal qualification testing

Software FQT will be conducted by C&C Integration and Test Group personnel. Figure 3.1.3-1 shows the relationship of the C&C Integration and Test group within the C&C Flight Software IPT. This group is independent from the C&C Design and Code group and performs software FQT prior to delivery of the NCS and CCS to the S/W Integ IPT. In addition, the C&C Integration and Test Group is responsible for preparation of the STP, STD, and STRs. It is also responsible for conduct of TRRs and support of System Integration Testing.

5.1.2 Personnel – formal qualification testing

The C&C Integration and Test group consists of a software test Responsible Engineer (RE) and software test engineers assigned to the C&C Flight Software IPT. The software RE is responsible for the development of the requirements and the test engineers are responsible for developing test cases. Formal testing will be performed by the test engineers and the software RE. Personnel requirements, skills, and staffing levels are maintained by the C&C Flight Software IPT manager in accordance with Prime Contractor requirements. Further definition of the personnel requirements and skill levels may be found in the C&C Flight Software IPT TEP.

5.2 Test approach/philosophy

Software FQT ends at CSCI testing. However, if limitations of the test environment at the CSCI testing level preclude the qualification of all requirements, then those requirements may be allocated to enditem and/or stage integration levels. The following paragraphs expand the approach and provide a philosophy that supports software FQT decisions and activities.

5.2.1 Approach

The test approach is in accordance with the guidelines of DOD-STD-2167A and all section 3 SRS functional and performance requirements will be explicitly covered, consistent with section 4 of the SRS. The CSCI testing is the FQT activity that follows dry run. The NCS and CCS CSCIs are each fully integrated prior to software FQT. The NCS and CCS CSCIs are then each qualified as stand-alone boxloads. Successful boxload qualification demonstrates that the NCS and CCS CSCIs are capable of performing as specified.

Testing is conducted in accordance with the planning in the STP and the test cases and procedures documented in the STD. The NCS and CCS CSCIs will each have their own documentation. The STP will define the scope, the type or classes, the level and identify each test case. The individual test case information will define the objective, level, type, cross reference to the requirement, and type of data to be collected. The STD will define each test case in detail, including the hardware and software

configurations, initial conditions, simulation conditions, test inputs, detailed test instructions and expected results.

The overall test results will be reported in the STR. The STR will also include a report of each of the test cases including any problem reports produced and deviations from the test procedure if any are required. The NCS and CCS CSCIs will each have their own STR.

5.2.2 FQT philosophy

The philosophy of software FQT supports a process that allows the C&DH Subsystem Provider IPT to determine whether software configuration items comply with the allocated requirements for that item.

CSCI requirements are found in SRSs and ICDs (Part 1). Because these requirements are allocated from higher-level specifications, it is possible that some requirements such as performance, timing, or external interface requirements cannot be totally verified in a pure software environment. In these instances it is permissible to allocate them to enditem or United States On-orbit Segment (USOS) testing where adequate hardware/software integration testing is supported.

The definition of CSCI qualification requirements (SRS section 4 and the STP) are used to define the qualification effort. CSCI Testing is complete when every SRS requirement has been exercised as defined in section 4 of the SRS and the STP.

CSCIs residing in their target processors, such as the NCS and CCS CSCIs, often will not be physically located with the enditem hardware which they functionally control. This causes software planning to take into account the hardware/software integration of CSCI firmware controllers remotely located on these enditems. Qualification will require exercising the enditem with hardware and software interfaces provided by simulations (or simulators) as required.

5.3 Test planning assumptions and constraints

5.3.1 Assumptions

The test planning effort is making assumptions when defining the tests which will be a part of the FQT. The assumptions considered during NCS and CCS CSCI test planning are:

a. PSPF is fully certified and under configuration control.

b. Versions of target software are available from configuration management as required.

c. Sufficient simulations are available to provide required test fidelity.

d. Ready access exists to the current requirements data base.

e. Appropriate supporting personnel are available to provide anomaly resolution.

f. On-board software loading is beyond the scope of the NCS and CCS CSCI testing activities.

g. GFE supplied software is qualified.

h. Flight software deliveries for testing analysis.

5.3.2 Constraints

The test planning is taking into account constrains which are present when defining the tests. The NCS and CCS CSCI qualification test planning is constrained by:

a. Capabilities of the PSPF

b. Level of testing to be accomplished

c. Fidelity and capabilities of existing simulations

d. Operation of the software under test in its target processor

e. Test conducted that neither modifies nor corrupts the software under test

6. SOFTWARE PRODUCT EVALUATIONS

Software Product Evaluations (SPEs) are a critical part of the software development process. Prior to the submittal of each C&C deliverable software item to the C&DH S/W Integ IPT, the C&C Software IPT will internally perform an examination, known as the SPE. The objective of the SPE is to ensure that the deliverable item is acceptable in terms of its ability to satisfy its requirements and that the C&C SSPS are complied with. A SPE is performed on deliverable products produced during each software life-cycle phase, as required by DOD-STD-2167A and modified by Table 6.0-1, SPE Occurrence by Phase.

SPEs will be held towards the end of each software development phase, following initial completion of the product and prior to its delivery to the customer. SPEs will be held during each software life-cycle phase, for at least those software items identified in Table 6.0-1, SPE Occurrence by Phase. Several evaluations of the same item may occur during the same or different phases. It is not the intent to force evaluators to repeat evaluations that have already occurred, unless required due to changes. Those conducting evaluations may use the results and reports from prior evaluations to verify compliance with requirements that have not been modified since the prior evaluation.

6.1 Organizations and Resources

6.1.1 Organizational Structure – Software Product Evaluations

Each software developer will be responsible for planning and conducting SPEs on assigned software products/items. SPE membership will include personnel with the various areas of expertise deemed necessary to verify the product. Membership is assigned through the C&C Software IPT and will include SQA and peer software developers. Software product evaluations are conducted prior to delivery of software products and are chaired by the responsible engineer.

TABLE 6.1.1-1 SPE OCCURRENCE BY PHASE

PRODUCT/ITEM	SOFTWARE LIFE CYCLE PHASE					
	Requirement Definition	Preliminary Design	Detailed Design	Code and CSU Testing	CSC Integration & Testing	CSCI Test
S/W Development Plan	SPE	-	-	-	-	-
S/W Requirements Spec.	SPE *	-	-	-	-	-
S/W Test Plan	-	SPE	-	-	-	-
STD Vol. 1 (Test Cases)	-	-	SPE	-	-	-
ICD Part 1	-	SPE*				
ICD Part 2	-	-	SPE*	-	-	-
Source Code	-	-	-	SPE	-	-
STD Vol. 2 (Test Proc.)	-	-	-	-	SPE	-
S/W Test Report	-	-	-	-	-	SPE
Version Description Doc.	-	-	-	-	-	SPE
Data Base Design Doc	-	-	SPE	-	-	-
Top Level Design Doc	-	-	SPE	-	-	-
S/W Standards and Procedures Spec	-	SPE	-	-	-	-
Software Users Manual	-	-	-	-	-	SPE

* See section 9.1 for further clarification

6.1.2 Personnel – Software Product Evaluations

The following roles and responsibilities have been identified for the SPE:

a. Review Leader – SPEs are chaired by the responsible engineer. The responsibilities of the team leader include coordinating and scheduling evaluations, distributing review materials, and preparing evaluation records.

b. Reader – SPEs have a reader, who walks the review team through the software product under review by paraphrasing the material in a logical, orderly manner.

c. Recorder – SPEs have a recorder who records defects identified in the software product. SQA personnel will function as the recorder, whenever feasible.

d. Reviewers – Reviewers are assigned technical experts with the appropriate skill mix to properly critique the software product being evaluated. Depending on the product, they would include system engineers, software engineers, peer developers, hardware engineers, validation and verification personnel, software configuration management, and software documentation experts. The reviewers identify defects in the software product and question the reader and author to ensure that a complete and common understanding of any area in question is attained. Their responsibilities also include ensuring that the evaluation criteria is met.

e. Author – The author is the developer of the software product under review; also known as the responsible engineer. The responsibility of the author is to answer questions about the software product, to resolve identified problems, and to implement the changes to correct identified defects.

6.2 Software product evaluation procedures

SPE procedures, planning guidelines, and tools will be in accordance with the software quality assurance sections of the Prime S&MA Plan. Exit criteria or checklist items for each SPE will be developed by the C&C Software IPT during the preliminary design phase. The conduct of the evaluations and the exit criteria or checklist items will be documented in the C&C SSPS. No special software tools have been identified for conduct of SPEs.

6.3 SPE records

SPE records will be maintained by the responsible engineer who is chairing the SPE. The records will also be included as part of the SDF which will be maintained for each software product. The records will include at least the following data:

a. Evaluation date.

b. Evaluation participants.

c. Evaluation purpose and criteria.

d. Evaluation results, including detected problems, with reference to the appropriate Software SPRs, as applicable.

e. Recommended corrective actions.

f. Completion status and next assigned activity, as appropriate.

6.4 Activity-dependent evaluation records

SPEs and reviews will be conducted during each software life cycle phase. SPEs are final examinations of deliverable software products as they are approaching submittal to the Prime. The minimum occurrences of SPEs and reviews by software life cycle phase is presented in Table 6.0-1, SPE Occurrence by Phase.

7. SOFTWARE CONFIGURATION MANAGEMENT

C&C software Configuration Management (CM) requirements shall be governed by SSP 41170, Configuration Management Requirements. Software configuration management is the responsibility of the C&DH Software CM IPT. This includes support to the various Program and configuration control boards, the software change process, the software libraries control processes, the software audit process, the software configuration identification process, the software status accounting process, and liaison with other elements of CM.

The detailed explanation of the CM processes and methods to be used on all segments of the ISS is contained in SSP50123, Configuration Management Handbook (CMH). The following sections relating to SCM describe the software-unique CM activities and refer to the appropriate sections in the Configuration Management Handbook for general, overall CM information.

7.1 Organization and Resources – Configuration Management

The SCM function is tiered below the C&DH Subsystem IPT level to lower-level AITs and IPTs. Each IPT leader is responsible for assuring that SCM plans and procedures are implemented and are in conformance with the policies and procedures of the CMH. SCM serves as the primary focal point for this activity in support of the IPT leaders.

The following sections describe the organization and resources necessary for SCM to support the C&C Software IPT.

7.1.1 Organizational Structure – Configuration Management

The NASA/Prime SCM is a part of the Vehicle AIT and Space Station AIT. SCM personnel are further assigned to the IPTs which have software development, build, and verification responsibilities.

7.1.2 Personnel – Configuration Management

The base staffing associated with SCM varies throughout the life-cycle and generally tracks the classic software development staffing profile. In addition, SCM in its review process role, employs SCM Specialists on a temporary basis to support major milestone reviews.

7.2 Configuration Identification

For each software item (CCS, NCS, CES or NES) configuration identification is established for software documents and code. The initial approved configuration identifications establish baselines from which subsequent changes are controlled. The configuration identifications and baselines to be established for the software products are defined as follows:

a. Allocated Baseline - The requirements (SRS and Part 1 ICD) are established at S/W PDR as the Allocated Baseline. They will reside on the Program Automated Library System (PALS) and will be controlled by the program change process defined in the CMH.

b. Developmental Configuration - the software and associated documents that defines the evolving configuration of the CSCI during development. The developmental configuration is under IBL control. The developmental configuration describes the software design and implementation. The developmental configuration for a software item consists of source code listings, related documents, and the changes that have accumulated since the last release of the CSCI. Changes are collected in the EDL for a software item until the change can be incorporated into a controlled baseline (the IBL).

c. Product Baseline - A baseline is defined as software and associated documents designated, fixed, and controlled at a specific point in each item's life cycle. This point in the life cycle is at the completion of the CSC Integration Test phase.

7.2.1 Developmental Configuration Identification

Developmental configurations which will be controlled during the life cycle of the CCS, NCS, CES and the NES software development are identified in section 3.2.1. These configurations include the following:

a. CCS Breadboard baseline - This baseline will be used to support initial integration of the Breadboard software. Basic capabilities to support the Brassboard development will be provided, including the capability to move data into and out of the MDM.

b. CCS Brassboard baseline - This baseline will be used to support integration of the Brassboard software. This version will include the functional capabilities to support the CSCI external interfaces.

c. CCS Proto-flight baseline - This baseline will be used to support integration of the Proto-flight software. This version will include all of the functional capabilities of the CSCI with the exception of some error handling capabilities.

d. CCS baseline - This baseline will be released at the completion of the design level integration test of the CSCI to support formal qualification testing of the CCS. This version will include all of the functional capabilities of the CSCI.

e. NCS Breadboard baseline - This baseline will be used to support initial integration of Breadboard software. Basic capabilities to support the Brassboard development will be provided, including the capability to move data into and out of the MDM.

f. NCS Brassboard baseline - This baseline will be used to support integration of the Brassboard software. This version will include the functional capabilities to support the CSCI external interfaces.

g. NCS Proto-flight baseline - This baseline will be used to support integration of the Proto-flight software. This version will include all of the functional capabilities of the CSCI with the exception of some error handling capabilities.

h. NCS baseline - This baseline will be released at the completion of the design level integration test of the CSCI to support formal qualification testing of the NCS. This version will include all of the functional capabilities of the CSCI.

i. CES Breadboard baseline - This baseline will be used to support initial integration of Breadboard software. Basic capabilities to support the Brassboard development will be provided, including the capability to move data into and out of the MDM.

j. CES Brassboard baseline - This baseline will be used to support integration of the Brassboard software. This version will include the functional capabilities to support the CSCI external interfaces.

k. CES Proto-sim baseline - This baseline will be used to support integration of the Proto-sim software. This version will include all of the functional capabilities of the CSCI with the exception of some error handling capabilities.

l. CES baseline - This baseline will be released at the completion of the design level integration test of the CSCI to support formal qualification testing of the CCS. This version will include all of the functional capabilities of the CSCI.

m. NES Breadboard baseline - This baseline will be used to support initial integration of Breadboard software. Basic capabilities to support the Brassboard development will be provided, including the capability to move data into and out of the MDM.

n. NES Brassboard baseline - This baseline will be used to support integration of the Brassboard software. This version will include the functional capabilities to support the CSCI external interfaces.

o. NES Proto-sim baseline - This baseline will be used to support integration of the Proto-sim software. This version will include all of the functional capabilities of the CSCI with the exception of some error handling capabilities.

p. NES baseline - This baseline will be released at the completion of the design level integration test of the CSCI to support formal qualification testing of the NCS. This version will include all of the functional capabilities of the CSCI.

7.2.2 Identification Methods

Each controlled software item will be assigned a unique, functionally descriptive name. Each configuration of each item released to the CPL will have a unique item version number. Additionally, each CSCI will be assigned a unique part number, which will also include a part version number. Each time the software item is revised and released, both the item version number and the part version number will be incremented.

7.3 Configuration Control

All proposed changes to the baseline to which NASA/Prime has control, will be properly documented, evaluated, coordinated, and dispositioned according to the procedures outlined in Appendices C & L of the Configuration Management Handbook.

7.3.1 Flow of Configuration Control

This section describes the process used to identify, analyze, correct, and review changes to a controlled software item or associated documents. The SPR form is used to identify and track problems found in software. SPRs can be generated as a result of discovery of software/system errors through test or operations, or via a program change. There is no restriction as to who can generate an SPR. Program Change Memos (PCMs) are used to identify and track problems in the SRSs and ICDs.

All changes to CPL controlled software items and software documents require an SPR to be written and approved before the change can be incorporated into a product. Figure 7.3.1-1, Corrective Action Process, depicts the process by which a SPRs are submitted, reviewed, dispositioned, and implemented including the activities and organizational responsibilities from SPR initiation through closure.

A summary of the corrective action process as depicted in Figure 7.3.1-1, Corrective Action Process, follows. When problems are identified, an SPR will be drafted and submitted to SCM, which serves as the processing focal point for proposed changes. SCM assigns a number to the SPR, enters the SPR in the SPR log, and makes copies available to Engineering and SQA for problem analysis and coordination. SCM compiles responses and provides the data to a C&C Software Review Board (SRB) for disposition. The SRB will either approve or disapprove the SPR. If the SPR was disapproved, SCM will close out the SPR and notify the originator of the SPR. Facility problems will be reported and resolved in accordance with the SDIL CM Handbook and M&O Standard Operating Procedures using the SDIL DR process. If the SPR was approved, the SRB will evaluate the SPR for cost impacts. If no cost impact, the SRB will process and implement the SPR. This includes developing and testing the code. If the SPR has a cost impact, the SRB will initiate the correct change paper and submit to SCM for tracking and coordination. Once the change paper is authorized, the SRB will process and implement the SPR. This includes developing and testing the code. Once the change has been implemented, SCM will close the SPR and notify the originator.

The development of the CCS SRS is internally controlled by the SMC IPT until PDR. After PDR, the SRS is updated and is placed under Prime control by releasing it through the NASA/Prime Engineering Release Unit (ERU). Following baselining, updates to the SRS are effected using the processes documented in Appendix C of SSP 50123, i.e.,

a. SRS only changes: Team Change Proposals (TCPs) and the Prime Baseline Change (Class II) process.

b. Changes which affect NASA controlled products: Engineering Change Proposals (ECPs) and the NASA Baseline Change (Class I) process.

7.3.2 Reporting Documentation

To correct problems, evaluate potential software changes, or make enhancements to software products under configuration control, an SPR must be submitted, reviewed and subsequently approved or disapproved. An example SPR form is shown in the Prime Contractor SDP. Details of the process for documenting problems on the SPR and for processing of the SPR will be defined in the C&C SSPS.

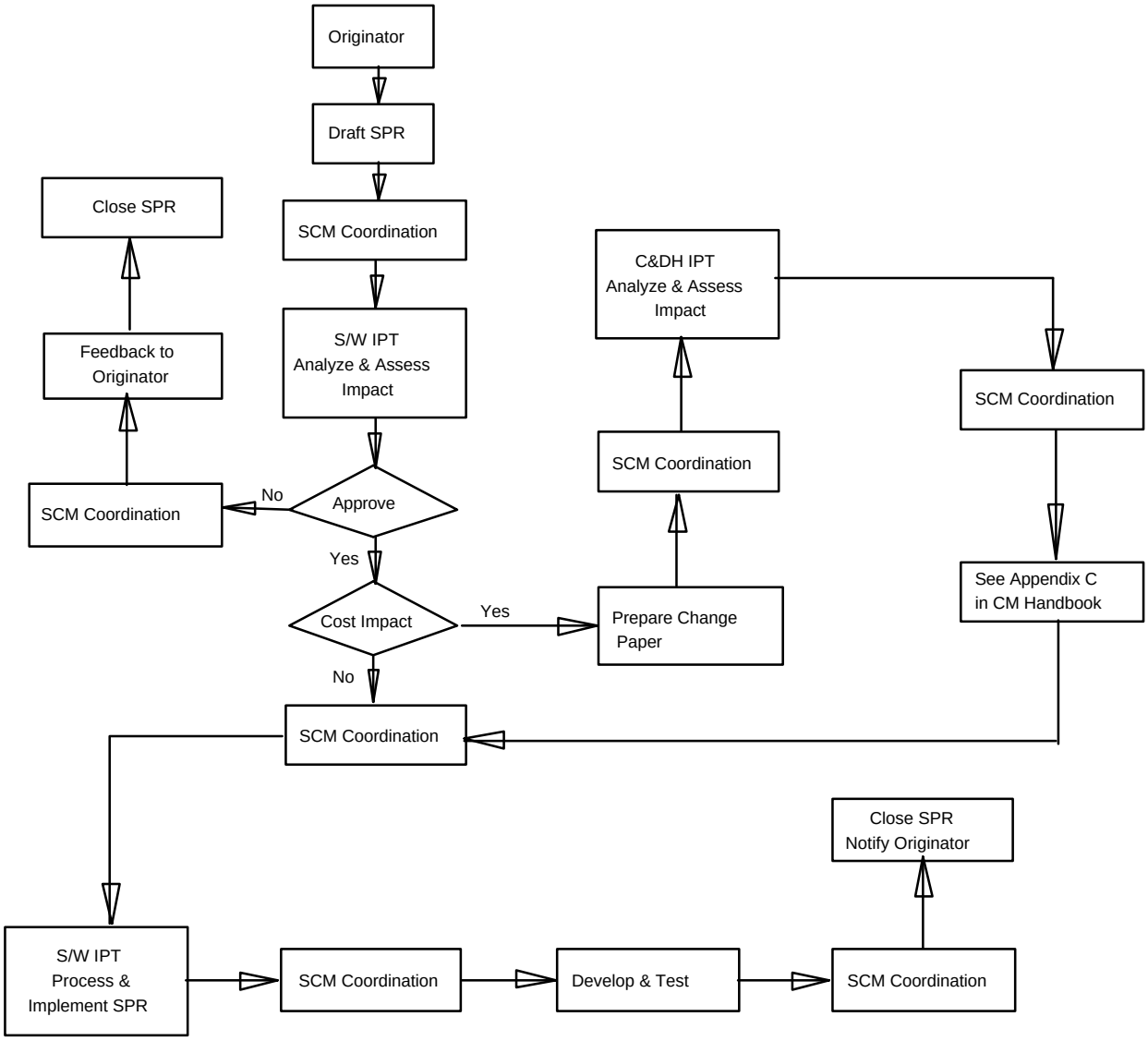


FIGURE 7.3.1-1 CORRECTIVE ACTION PROCESS

7.3.3 Review Procedures

The C&C Software IPT, which includes members from SCM and SQA, is the governing review board for controlling software configuration management, as described in the preceding sections. No separate review board is associated with software configuration management.

7.3.4 Storage, Handling, and Delivery of Project Media

Released software items will be stored on magnetic media and controlled by SCM. User copies of software items to be delivered for test or operations will be copied from SCM master copies and delivered to the user by SCM.

7.3.5 Additional Control

None.

7.4 Configuration Status Accounting

Configuration status accounting provides the information needed to identify the configuration and determine the status of change proposals and deviations and waivers, including implementation status. The status accounting report for a CSCI provides information concerning traceability of configuration baselines and changes to them, computer programs, specific CSCIs and their lower level components, and related documentation in a controlled access environment. The technique for tracking and reporting the configuration status of CSCIs for the Prime Contractor is the same for hardware, and may be integrated into a single reporting system.

Configuration status accounting records and reports assure that there will be a configuration record documenting all approved configuration changes to all CSCIs. Periodic status reports will be provided on all products in the developmental configuration and the allocated and product baselines. These reports have the following objectives:

a. Provide traceability of changes to controlled products;

b. Serve as a basis for communicating the status of each CSCI and its released components;

c. Serve as a vehicle for ensuring that delivered documents describe and represent the associated software.

Two configuration status accounting reports are the Configuration Identification Index (CII) and the Configuration Status Accounting Report (CSAR). These records document CSCI status from initial design acceptance until program completion; thus a current account of approved CSCIs, approved changes to CSCIs, and actual versus approved configuration is maintained. Software configuration status accounting also tracks the status of software changes and enhancements. This information will be entered into the CII and CSAR, where appropriate, when this information is maintained separately. SQA will audit configuration status accounting records to verify their accuracy accounting of configuration status. The status accounting function will include, at a minimum:

a. Current schedule

b. Progress indication

c. Summary of current changes including:

    1. Reason for change

    2. Impact of change

d. Summary of current SDR status

e. Definition of key milestones and deliverables

7.5 Configuration Audits

As part of the software life-cycle, audits of software are scheduled between software CDR and delivery. FCA and PCA are held following completion of FQT. Both CM and Software Quality Assurance participate in audits to assure they are in conformance with requirements.

7.6 Preparation for Specification Authentication

Software specifications (SRS) for the ISS program will be authenticated in accordance with the Configuration Management Handbook. Specification changes after authentication by the Prime Contractor will be documented and processed as described in the Configuration Management Handbook.

7.7 Configuration Management Major Milestones

Software products will be evaluated at the milestone reviews specified in section 3.2 of this plan. Configuration baseline establishment associated with the successful completion of these milestones is defined in the Configuration Management Handbook.

This page intentionally left blank.

8. OTHER SOFTWARE DEVELOPMENT FUNCTIONS

This section intentionally left blank.

This page intentionally left blank.

9. NOTES

9.1 Exceptions to the Prime Contractor Software Development Plan

The C&C IPT software is a critical component to the success of the ISS program. As such the design and external interfaces for the software will need to be clearly communicated to the many organizations that need to understand its functions. DOD-STD-2167 specifies two documents, a TLDD and a DBDD, which allow for more concise communication of the high level design and the external interface data definitions than can be accomplished with the Software Product Specification (SPS). The remaining detailed design information will be documented in the Software Development Folders (SDFs) per the Prime Contractor SSPS.

Exception is taken to the requirement for an STD Volume 1 as input and success criteria for CDR. A FQT TIM will be held separately for the review of the STD Volume 1. The TIM will be held by the C&C Software Integration and Verification Group which is part of the C&DH Verification IPT.

Exception is taken to the requirements for SPEs on the SRSs as described in Section 6 of the Prime SDP and Section 6 of this plan. These products are reviewed in their respective SSRs in accordance with DOD STD-2167A, MIL-STD-1521B and the Prime SDP; however, SPEs were not accomplished prior to these reviews in the formalized manner described in this plan. However, their development has involved iterative and ongoing self- and external organization- evaluation which meet the intent of the SPE process without its rigor. Further, part of the planned requirement development process includes informal requirements ‘walkthroughs’, scheduled and conducted by the SMC AIT, between the requirements authors, C&DH architects, and corresponding C&C IPT designer(s). The intent of these ‘walkthroughs’ is to ensure a common understanding of required functionality among the development community. The output of these reviews will include SRS Redlines and issues for forward work, much as the SPE Process describes. Records and results of these ‘walkthroughs’ are retained in the SDF itself or with a pointer to the location of the information.

Although reviews were conducted, formal SPEs have not been performed for all software products for NCS R1. The informal SPEs produced informal documentation for SDP, STP, STDs, ICDs, Source Code, DBDD, TLDD, STR, VDD, SSPS, and Sum. That documentation will be located within the CSCI Index SDF itself or with a pointer to location of the information. However, formal SPEs as defined in Sect 6 of the SDP will be performed for subsequent software products (NCS R2: STR and CCS R1: STD, STR and SUM) as defined and tailored within the SDP and SSPS.

9.1.1 Programming Language

Appendix A of the Prime SDP identifies the software language requirements associated with the flight software development. The language identified is Ada for NCS and CCS. Assembly language is used for a small portion of the flight software development. The use of assembly language is limited to simple data move, data collect, checksum and memory clear/set utilities. A benchmark test was performed early on to determine the performance characteristics of data move procedures with Ada. The timing results gained from this benchmark dictated the use of assembly language in order to meet performance requirements.

It is recognized that assembly language is not a self documenting language. This can result in maintenance being more costly and it can be more difficult to locate and correct coding errors. For this reason assembly language is only used when absolutely required and for routines that are very small and are not likely to change during the life of the program. The use of assembly language is limited to less than 1% of the delivered flight software.

9.2 Exceptions to the Prime Contractor Software Standards and Procedures Specification

SDFs for the C&C IPT flight software CSCI are created for two levels of hierarchical decomposition. These levels are the CSCI and one level below referred to as TLCSC. No SDFs are developed for lower level decomposition. Both CCS and NCS will follow the same format as defined within this exception.

SDF templates which are compliant with the C&C IPT as defined in the SSPS and tailored by these exceptions have been created for NCS and CCS and are located on the server for the designers to populate with functionally specific data.

The Index CSCI SDF will contain the sections as outlined in the C&C SSPS (SDIL-006) paragraph 4.1.1 with the exception of the path name to the SPS, which has been replaced with a DBDD and TLDD per paragraph 9.1, above.

Each second level SDF will contain the items of Section 4.1.2 and the items of Section 4.1.3 of the C&C SSPS (SDIL-006) with redundancies between the paragraphs 4.1.2 and 4.1.3 removed. Redundancies between the Index SDF and second level SDF will be removed

9.2.1 CSCI SDF Format

The CSCI Index SDF will vary from the Prime SSPS in the following areas:

The Activity Log will contain the release dates for each of the software product releases.

In the Activity Log, the Preliminary Design (activity), Preliminary Design Peer Review (event), and Preliminary Design Review (event) sections will be replaced with a section called Design Activity.

The Activity Log will not contain :

- **Requirements (activity)**
- **SRS Peer Review (event)**
- **SSR (event)**
- **Preliminary Design (activity)**
- **Preliminary Design Peer Review (event)**
- **Preliminary Design Review (event)**
- **Detail Design (activity)**

- **Detail Design Peer Review (event)**
- **Critical Design Review (event)**
- **Unit Test Preparation (activity)**
- **Coding (activity)**
- **Peer Code Review (event)**
- **Unit Testing (activity)**
- **Unit Test Report Preparation (activity)**
- **CSC Integration Testing (activity)**

9.2.2 Second Level SDF Format

The Second Level SDFs will vary from the Prime SSPS in the following areas:

The redundancy will be removed from the SDF by not repeating what is contained in the Index SDF (path names to STP, STD, STR (SPS, see above paragraph), VDD, Schedule and Status).

The "Test Cases" and "Test Procedures" sections are combined and named as a "Unit Test Plan" section for unit test cases/procedures and a "Integration Test Plan" section for integration test cases/procedures.

The "Test Results" section is named as a "Unit Test Results" section for unit tests, and a "Integration Test Results" section for integration test results.

The "Source Code for stubs and drivers" section are contained in a section called "Unit Test Completion Matrix" for unit test and a section called "Integration Test Completion Matrix" for Integration Test.

The Activity Log will contain :

- **Preliminary Design (activity)**
- **Preliminary Design Review (event)**
- **Detail Design (activity)**
- **Critical Design Review (event)**
- **Unit Test preparation (activity)**
- **Coding (activity)**
- **Peer Code Review (event)**

- **Unit Testing (activity)**
- **CSC Integration Testing (activity)**
- **Release Date(s)**

The Activity Log will NOT contain the following :

- **Requirements (activity)**
- **SRS Peer Review (event)**
- **SSR (event)**
- **Preliminary Design Peer Review (event)**
- **Detail Design Peer Review (event)**
- **Unit Test Report Preparation (activity)**

a. Content of Software Design paragraph is included. Referenced "Detailed Design" information is available in the "Notes" portion of the SDF via included documentation or pointers to this information.

b. The "Review Notes" (TLCSC SDFs only) section is replaced with a "Reviewer Comments" section.

c. The "Test Log" section is named as an "Informal Test" section.

9.3 "Grandfathered" Software from Space Station Freedom Program

None.

9.4 Acronyms and Glossary

AIS	Automated Information System
AIT	Analysis and Integration Team
AL	Air Lock
APM	Attached Pressurized Module
C&C	Command and Control
C&DH	Command and Data Handling
CCS	Command and Control Software
CDR	Critical Design Review
CES	Command and Control Environment Simulation
Cf	Consequence factor
CII	Configuration Identification Index
CM	Configuration Management

CMH	Configuration Management Handbook
CMS	Code Management System
CPL	Central Program Library
CPU	Central Processing Unit
CSA	Canadian Space Agency
CSAR	Configuration Status Accounting Report
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DBDD	Data Base Design Document
DEC	Digital Equipment Corporation
DID	Data Item Description
DOD	Department of Defense
DR	Data Requirement
ECLSS	Environmental Control and Life Support System
ECP	Engineering Change Proposal
EDL	Engineering Design Library
ERU	Engineering Release Unit
ESA	European Space Agency
FAR	Federal Acquisition Regulation
FCA	Functional Configuration Audit
FDIR	Fault Detection, Isolation, and Recovery
FEU	Functionally Equivalent Unit
FQT	Formal Qualification Test
FRR	Flight Readiness Review
FTP	File Transfer Protocol
GFE	Government Furnished Equipment
GN&C	Guidance Navigation and Control
GSE	Ground Support Equipment
HAB	Habitation
IBL	Integration Build Library
ICD	Interface Control Document
ICWG	Interface Control Working Group
IP	International Partner Internet Protocol
IPR	In-Process Review
IPT	Integrated Product Team
ISI	Integrated Systems Incorporated
ISS	International Space Station
IV&V	Independent Verification and Validation

JEM	Japanese Experiment Module
JSC	Johnson Space Center
M&O	Maintenance and Operation
MATE	MDM Applications Test Environment
MBF	Mission Build Facility
MDM	Multiplexer/Demultiplexer
MIL	Military
NASA	National Aeronautics and Space Administration
NASDA	National Space Development Agency of Japan
NCS	Node 1 Control Software
NES	Node 1 Environment Simulation
ORU	Orbital Replaceable Unit
PALS	Program Automated Library System
PC	Personal Computer
PCA	Physical Configuration Audit
PCM	Program Change Memo
PDR	Preliminary Design Review
Pf	Probability factor
PG	Product Group
PSPF	Prime Software Production Facility
PTR	Port Thermal Radiator
PVCA	Photovoltaic Controller Application
PVCU	Photovoltaic Controller Unit
RTE	Run-Time Environment
RE	Responsible Engineer
RF	Risk Factor
RMP	Risk Management Plan
RS	Recommended Standard
RSA	Russian Space Agency
RTU	Request To Update
RT	Remote Terminal
S&MA	Safety and Mission Assurance
SCSI	Small-Computer System Interface
SCM	Software Configuration Management
SCTF	Sonny Carter Training Facility
SDD	Software Design Document
SDF	Software Development Folder
SDIL	Software Development and Integration Laboratory
SDL	Software Development Library
SDP	Software Development Plan
SDR	System Design Review

SE	Systems Engineering
SEE	Software Engineering Environment
SLOC	Software Lines of Code
SMC	Station Management and Control
SPE	Software Product Evaluation
SPR	Software Problem Report
SPS	Software Product Specification
SQA	Software Quality Assurance
SRB	Software Review Board
SRR	System Requirements Review
SRS	Software Requirements Specification
SSPS	Software Standards and Procedures Specification
SSR	Software Specification Review
STD	Software Test Description Standard
STP	Software Test Plan
STR	Software Test Report
SUM	Software User's Manual
SVF	Software Verification Facility
SW	Software
TBD	To Be Determined
TCP	Team Change Protocol Transmission Control Protocol
TCS	Thermal Control System
TEP	Team Execution Plan
TLCSC	Top Level Computer Software Component
TLDD	Top Level Design Document
TPM	Technical Performance Measure
TRR	Test Readiness Review
TSE	Test Support Equipment
UDP	User Datagram Protocol
USL	United States Laboratory
USOS	United States On-orbit Segment
VAX	Virtual Address Extension
VDD	Version Description Document
VMS	Virtual Memory System

Central Software Library

The central electronic repository of flight software and data which have been delivered to the Mission Build Facility for integration, test, and incorporation into one or more mission flight builds.

Computer Program

A sequence of coded instructions which encode a thought process or algorithm that may be executed by a computer system.

Deliverable Software

Software, including flight software, ground support software, support and development software, simulation software, and test software, that is to be delivered to the Prime.

Flight Software

Body of operational software on the ISS at a given point in time and applies to all flight software CSCIs developed for use with firmware controllers, MDMs, sensors, effectors, and other ISS defined flight Orbital Replaceable Units (ORUs). For qualification and acceptance tests of hardware using flight software, the software may be modified to the extent necessary to conduct the qualification and acceptance tests.

Formal Qualification Test

Process that allows the Prime to determine whether a software configuration item complies with the allocated requirements for that item.

Ground Support Equipment (GSE)

Contract-deliverable equipment (hardware/software) used on the ground to test, transport, access, handle, maintain, measure, verify, service, and protect flight hardware/software.

Ground Support Equipment Software

Software that is contained in Ground Support Equipment.

Module

The lowest level software design/implementation unit (i.e., Unit or CSU, a subdivision of a CSC).

Non-deliverable Software

Software that is developed and used within a Tier 1 Subcontractor and is not being delivered to the Prime.

Non developmental Software

Includes reusable software, commercially-available software, and government-furnished software.

Non-flight Software

Software used to support ground activities including ISS flight software design, development, integration, and verification; flight article and enditem design, development, and qualification; on-orbit stage configuration integration and verification; and launch package integration. Categories include test software, including simulations; SVF software; GSE/Test Support Equipment (TSE) software; and ground software, including MBF software.

Requirements Analysis

Software Requirements Analysis is a specific term which applies to specific processes which are performed after the software requirements are defined and is the key to an effective software design. It is used in both the Structured Analysis and Object-Oriented development approaches.

Reusable Software

Software that is reusable which provides low-risk, low-cost to meeting ISS software requirements. Reusable Software must meet all process requirements and all data items associated with this software must meet format and content standards for the ISS software.

Simulation

Provides the environment for the integration and verification of the flight software and avionics from software development through integration. Simulations include the test environment with enditem and segment simulations, environmental simulations which represent the ISS on-orbit environment and dynamics, and sensor/effector simulations.

Software Product Evaluations

Evaluations that are performed on deliverable products produced during each product development phase to ensure compliance to requirements.

Software Verification Facility

An ISS program facility comprised of multiple test strings of functionally equivalent MDMs connected with test unique hardware and software that will support flight software stage verification and validation, operational procedure validation.

Support Software

Categories of support software include facility, test, GSE/TSE and ground software.

Test Software

Types of test software include vertical simulations, horizontal simulations, test software, and test configured flight software.

Test Support Equipment

Equipment that is designed for use by the contractor to support development, production, and test activities associated with the ISS flight hardware and software.

Test Support Equipment Software

Software that is contained in Test Support Equipment.

Validation

Ensures that each product reflects an accurate interpretation and execution of requirements and meets a level of functionality and performance that is acceptable to users.

Verification

Ensures that products comply with the specification requirements imposed on them.